Intelligent Systems and Applications

Proceedings of the 2020 Intelligent Systems Conference (IntelliSys) Volume 1

Springer

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-030-55179-7 ISBN 978-3-030-55180-3 (eBook)
https://doi.org/10.1007/978-3-030-55180-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Editor's Preface

This book contains the scientific contributions included in the program of the Intelligent Systems Conference (IntelliSys) 2020, which was held during September 3–4, 2020, as a virtual conference. The Intelligent Systems Conference is a prestigious annual conference on areas of intelligent systems and artificial intelligence and their applications to the real world.

This conference not only presented state-of-the-art methods and valuable experience from researchers in the related research areas, but also provided the audience with a vision of further development in the fields. We have gathered a multi-disciplinary group of contributions from both research and practice to discuss the ways how intelligent systems are today architectured, modeled, constructed, tested and applied in various domains. The aim was to further increase the body of knowledge in this specific area by providing a forum to exchange ideas and discuss results.

The program committee of IntelliSys 2020 represented 25 countries, and authors submitted 545 papers from 50+ countries. This certainly attests to the widespread, international importance of the theme of the conference. Each paper was reviewed on the basis of originality, novelty and rigorousness. After the reviews, 214 were accepted for presentation, out of which 177 papers are finally being published in the proceedings.

The conference would truly not function without the contributions and support received from authors, participants, keynote speakers, program committee members, session chairs, organizing committee members, steering committee members and others in their various roles. Their valuable support, suggestions, dedicated commitment and hard work have made the IntelliSys 2020 successful. We warmly thank and greatly appreciate the contributions, and we kindly invite all to continue to contribute to future IntelliSys conferences.

It has been a great honor to serve as the General Chair for the IntelliSys 2020 and to work with the conference team. We believe this event will certainly help further disseminate new ideas and inspire more international collaborations.

Kind Regards,

Kohei Arai
Conference Chair

Contents

Data-Driven and Artificial Intelligence (AI) Approach for Modelling and Analyzing Healthcare Security Practice: A Systematic Review

Prosper Kandabongee Yeng[1(✉)], Livinus Obiora Nweke[1], Ashenafi Zebene Woldaregay[2], Bian Yang[1], and Einar Arthur Snekkenes[1]

[1] Norwegian University of Science and Technology, Technolgien 22, 2815 Gjøvik, Norway
Prosper.yeng@ntnu.no

[2] University of Tromsø, The Arctic University of Norway, Hansine Hansens veg 18, 9019 Tromsø, Norway

Abstract. Data breaches in healthcare continue to grow exponentially, calling for a rethinking into better approaches of security measures towards mitigating the menace. Traditional approaches including technological measures, have significantly contributed to mitigating data breaches but what is still lacking is the development of the "human firewall," which is the conscious care security practices of the insiders. As a result, the healthcare security practice analysis, modeling and incentivization project (HSPAMI) is geared towards analyzing healthcare staffs' security practices in various scenarios including big data. The intention is to determine the gap between staffs' security practices and required security practices for incentivization measures. To address the state-of-the art, a systematic review was conducted to pinpoint appropriate AI methods and data sources that can be used for effective studies. Out of about 130 articles, which were initially identified in the context of human-generated healthcare data for security measures in healthcare, 15 articles were found to meet the inclusion and exclusion criteria. A thorough assessment and analysis of the included article reveals that, KNN, Bayesian Network and Decision Trees (C4.5) algorithms were mostly applied on Electronic Health Records (EHR) Logs and Network logs with varying input features of healthcare staffs' security practices. What was found challenging is the performance scores of these algorithms which were not sufficiently outlined in the existing studies.

Keywords: Artificial intelligence · Machine learning · Healthcare · Security practice

1 Introduction

The enormous increase in data breaches within healthcare is frightening and has become a source of worry for many stakeholders such as healthcare providers, patients, national and international bodies. In 2018, the healthcare sector recorded about 15 million records

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 1–18, 2021.
https://doi.org/10.1007/978-3-030-55180-3_1

which were compromised in about 503 data breaches [1, 2]. This was a triple of 2017 data breaches in healthcare [1, 2]. In the middle of 2019, the number of compromised records in healthcare were more than 25 million, implying that by the end of 2019, the number of compromised records might have sky rocketed [2]. Greater proportion of the breaches (59%) were perpetrated by insiders [1] who are authenticated users of the systems [3]. Most of the adversaries were motivated by financial gains (83%) and other motives such as convenience (3%), grudges (3%), industrial espionage (2%) [1]. The number of data breaches in healthcare has substantially exceeded that of the financial sector and almost caught up with other public sector entities [1].

The tremendous increase in data breaches in recent time within healthcare, have therefore left many to ponder about the possible causes. The healthcare data is comparatively richer and has become "honey-port", thereby attracting malicious actors [4, 5]. Health data has vast scientific, societal, and commercial values, which cause cyberattacks and black market targeting of this data. Healthcare data can be used to commit multiple dark activities in the dark web as detection of breaches, related updates and correction of the compromised data takes a longer time. Another angle of thought is that, the healthcare personnel are busy with their core healthcare duties and are less experienced in information security conscious care behavior. This leaves room for adversarial attacks. The technological measures (such as firewall, intrusion detection or prevention systems, antiviruses and security governance configurations) have been strengthen [6] and making it difficult for external cyber criminals to inappropriately access data [7, 8]. But there is no related development of "the human firewall" [9]. The human firewall is the information security conscious care behavior of the insiders [9, 10]. The human firewall has not gained equal attention, and this is the vulnerability which cyber criminals tend to exploit for easy entry [11]. By virtue of their access privileges, healthcare insiders are "double-edged sword". While their privileges enable them to provide therapeutic care to patients, healthcare staffs' errors and deliberate actions can compromise the confidentiality, integrity and availability (CIA) of healthcare data. Additionally, an attacker can masquerade as insiders to compromise healthcare data through various ways, including social engineering methods [3].

Furthermore, the healthcare environment is relatively complex and delicate, making it hard for healthcare information security professionals to design stricter access control policies. So, access control mechanisms in healthcare are mostly designed with a degree of flexibility to enable efficient patient management. While such design considerations are very important and meets the availability attribute of the CIA, the healthcare systems remain vulnerable. The broad range of access flexibility can be abused by the insiders. This can also be a dream for cyber criminals to adopt various diabolic means of gaining insiders credentials to enable them to equally have larger access. The incidence of data breaches could bring various consequences including denial of service for timely medical services, negative impact on mutual thrust between patient and healthcare providers, breaches to individual's privacy and huge finds to healthcare providers by national and international regulatory bodies.

The general objective of this study was to therefore to identify, assess, and analyze the state-of-the-art in artificial intelligence strategies and their hybrid aspects which can be used to efficiently detect anomaly and malicious events in healthcare staff's security

practices in their access related data towards improving counter-measures for healthcare staffs related security breaches.

Specific objectives include:

- Identifying AI learning algorithms which can be used to efficiently profile healthcare staff security practices, for anomalies detection.
- Assess and analyzed the design considerations of the methods (such as the tolerance ranges or thresholding mechanisms provided to accommodate non-treacherous user behaviors i.e., new users, mistakes and during emergencies) towards mitigating false positives.
- Assess and analyze their performance metrics and other suitable evaluation methods
- Determine associated challenges in the usage of the algorithms and how these challenges can possibly be overcome.

1.1 Motivation, Scope and Problem Specification

Healthcare Security Practice Analysis, Modelling and Incentivization (HSPAMI), is an ongoing research project in which an aspect involves modelling and analyzing data with AI methods to determine the security practices of healthcare staffs, towards improving their security conscious care behavior. In analyzing healthcare related data, there is the need to consider details of the methods and data sources in view of the unique and critical nature of the sector. In a related study, Walker-Roberts et al., conducted a systematic review of "the availability and efficacy of countermeasures to internal threats in healthcare critical infrastructure" [12]. Among various teams few machine learning methods were identified to be used for intrusion detections and preventions. The methods that were identified are Petri net, Fuzzy logic, K-NN, K-Decision tree (RADISH system) [12–14] and inductive machine learning methods [12, 13, 15]. In a similar way, Islam et al conducted a systematic review on data mining for healthcare analytics [16]. Categories such as healthcare sub-areas, data mining techniques, type of analytics, data and data sources were considered in the study. Most of the data analysis were for clinical and administrative decision making. The data sources were mostly human generated from electronic health records. Other studies which explored for related methods includes [17] and [18].

Even though, the studies [12, 16] were in healthcare context, details of the algorithms and data sources were not considered. For instance, the features of the data sources and algorithm performance methods, were not deeply assessed in their studies. Additionally, the studies of [17] and [18] were general and not healthcare specific. So unique challenges within healthcare environment were not considered in their study. To this end, the study aimed to explore in detail, AI methods and data sources in healthcare that can be efficiently used for modeling and analyzing healthcare professionals' behavior. Healthcare professionals and healthcare staffs were used interchangeably in this study to include but not limited to nurses, physicians, laboratory staff and pharmacies who access patients' records for therapeutic reasons.

2 Background

Security practice of healthcare staffs includes how healthcare professionals respond to the security controls and measures towards achieving the CIA goals of the healthcare organizations. Healthcare professionals are required to conduct their work activities in a security conscious manner to maintain the CIA of healthcare environment. For instance, borrowing of access credentials could jeopardize the purpose of access control for authorized users and legitimate accesses. Additionally, the inability to understand social engineering scammers' behavior can lead to healthcare data breaches.

Various ways can be adopted to observe, model and analyze healthcare professionals' security practices. Perception and socio-cultural context can be adopted by analyzing the healthcare staffs' security perception, social, cultural and socio-demographic characteristics with their required security practices. Also, Attack-Defense simulation can be used to measure how healthcare staffs understand social engineering related tricks. Furthermore, data-driven approach with artificial intelligence (AI) methods could be adopted to understand the security risk of each healthcare professions. The findings can then help decision makers to introduce appropriate incentive methods and solve issues which are hindering sound information security practice towards enhancing conscious care behavior. But this study is focused on exploring for appropriate AI methods and data sources that can be used to modeled and analyzed healthcare security practices. Therefore, psycho-socio-cultural context and attack-defense simulations are beyond the scope of this paper.

2.1 Data-Driven and Artificial Intelligence in Healthcare Security Practice Analysis

Advances in computational and data sciences along with engineering innovations in medical devices have prompted the need for the application of AI in the healthcare sector [19]. This has the potential of improving care delivery and revolutionizing the healthcare industry. AI can be referred to as the use of complex algorithms and software to imitate human cognitive functions [20]. It involves the application of computer algorithms in the process of extracting meaning from complicated data and to make intelligent decisions without direct human input. AI is increasingly impacting every aspects of our lives and the healthcare sector is not an exception. In recent years, the healthcare sector is experiencing massive deployments of AI in the bid to improve the overall healthcare delivery. There is currently no consensus on the classification of the applications of AI in healthcare. However, we rely on the classification of the application of AI in healthcare described in [21] to briefly discuss deployment of AI in healthcare.

The deployment of AI in healthcare sector has been classified in [21] to include; expert systems, machine learning, natural language processing, automated planning and scheduling, and image and signal processing. Expert systems are AI programs that have been trained with real cases to execute complicated tasks [22]. Machine learning employs algorithms to identify patterns in data and learn from them and its applications can be grouped into three, namely; supervised learning, unsupervised learning, and reinforcement learning [21]. Natural language processing facilitates the use of AI to determine the meaning of a text by using algorithm to identify key words and phrases in

natural language [21]. For automated planning and scheduling, it is an emerging field in the use of AI in healthcare that is concerned with the organization and prioritization of the necessary activities in order to obtain desired aim [21]. And image and signal processing involve the use of AI to train information extracted from a physical occurrence (images and signals) [21].

The common characteristics of all these applications is the utilization of massive data that is being generated in the healthcare sector to make better informed decisions. For instance, the collection of healthcare staffs' generated data, has been used for disease surveillance, decision support systems, detecting fraud and enhancing privacy and security [23]. In fact, the code of conduct for healthcare sector of Norway require the appropriate storage and protection of access logs of healthcare information systems for security reasons [24]. The healthcare staffs' accesses within the network or electronic health records (EHR), leaves traces of their activities which can be logged and reconstructed to form their unique profiles [24]. The healthcare staffs' accesses within the network or electronic health records (EHR), leaves traces of their activities which can be logged and reconstructed to form their unique profiles [25]. So, the appropriate AI methods can then be used to mine in such logs to determine the unique security practices of the healthcare staffs. Such findings can support management to adopt to the suitable incentivization methods towards improving on the security conscious care behavior in healthcare. Therefore, this study aims to explore for the appropriate AI methods and data sources that can be used to observe, model and analyzed the security practices of healthcare staffs.

3 Method

The objective of this study was to identify, assess and analyze the state-of-the-art data-driven and artificial intelligence (AI) algorithms along with their design strategies, and challenges. The study is towards analyzing healthcare professionals' security practices in the context of big data or human generated data in Healthcare Security Practice Analysis, Modeling and Incentivization (HSPAMI) project.

A literature search was conducted between June 2019 and December 2019 through Google Scholar Science Direct and Elsevier, IEEE Explore, ACM Digital. Different key words such as "Healthcare", "staff", "employee", "Information security", "behavior", "Practice", "Threat", "Anomaly detection", "Intrusion detection", "Artificial Intelligence" and "Machine Learning", were used. For a good quality searching approach, the key words were combined using Boolean functions of 'AND', 'OR' and 'NOT'. Peer reviewed journals and articles were considered. The inclusions and exclusions criteria were developed based on the objective of the study and through rigorous discussions among the authors. Basic selection was done by initially skimming through the titles, abstracts and keywords to retrieve records which were in line with the inclusion and exclusion criteria. Duplicates were filtered out and articles, which seems relevant, based on the inclusion and exclusion criteria, were fully read and evaluated. Other appropriate articles were also retrieved using the reference list of accepted literatures. Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) flow diagram was used to report the article selection and screening [26].

3.1 Inclusion and Exclusion Criteria

For an article to be included in the review, the study has to be an anomaly detection or intrusion detection in healthcare using artificial intelligence methods in healthcare professionals' generated access logs data or patterns. Any other article outside the above stated scope (such as articles in medical cyber-physical devices, body area networks etc.) including literatures in other languages, except English, were excluded.

3.2 Data Collection and Categorization

The data collection and categorization were developed based on the objective and through literature reviews and authors discussions. The categories have been defined exclusively to assess, analyzed and evaluate the study as follows:

Type of AI Method: This category includes explicit machine learning methods such as, Support Vector Machine (SVM), Bayesian network, etc.

Type of Input: This category includes the features which were used by the algorithm. This could include access location, time, log in failed attempts, etc.

Input Sources: This attribute refers to the kind of access logs data, which was used in the study. Such sources include browser history, network logs, host-based activity logs and electronic health records logs.

Data Format, Type, Size, and Data Source: This category could include file format such as XML, CSV.

Input Preprocessing: Defines how the data was preprocessed from unstructured to structured, and how missing and corrupted input data were handled.

Application Scenario: This category defines the context of which the algorithm was implemented such as intrusion or anomaly detection.

Ground Truth: Refers to the kind of training set used in training the model.

Privacy Approach: This defines the privacy method used to safeguard the privacy right of individuals who contributed to the data source.

Performance Metrics or Evaluation Criteria: This includes the measures used to assess the accuracy of the study. It includes metrics such as specificity, sensitivity, receiver operating characteristic (ROC) curves, and others.

Nature of Data Sources: This category specifies if the data used was synthetic or real data.

3.3 Literature Evaluation and Analysis

The selected articles were assessed, analyzed and evaluated, based on the above defined categories. The analysis was performed on each of the categories (Type of AI method, type of input, input source, preprocessing, learning techniques, performance methods, etc.) to evaluate the state-of-the-art approaches. Percentages of the attributes of the categories were calculated based on the total number of counts (n) of each type of the attribute. Some studies used multiple categories, therefore, the number of counts of these categories exceeded the total number of articles of these systems presented in the study.

4 Results

After searching in the various online databases, a total of 130 records were initially identified by following the guidelines of the inclusion and exclusion criteria in the reading of titles, abstracts, and keywords. A further assessment of these articles through skimming of the objective, method and conclusion sections led to a further exclusion of 77 articles which did not meet the defined inclusion criteria. After removing duplicates, 42 articles were fully read and judged. After the full text reading, a total of 15 articles were included in the study and analysis as shown in the Fig. 1. As shown in the Fig. 2 and 3, the topic of data-driven and AI for analyzing healthcare security practice has seen consistent interest.

As shown in Fig. 2, most of the literatures were identified in Google scholar and followed by IEEE Explore and ACM Digital Library.

The articles were published between 2010 and 2019 as shown in Fig. 3.

4.1 Evaluation and Analysis

Evaluation and analysis of the articles were carried out as described above, and the main finds are presented below.

Articles in the Study

The articles and their related categorizations, such as algorithms, features and data sources are shown in Table 1.

I. **Algorithms**

The algorithms which were found in the review are as shown in Table 2. KNN method was mostly used (17%), followed by Bayesian Network (14%) and C4.5 decision tree (10%).

II. **Features**

With reference to Table 3, the features which were mostly used include Users ID (19%), Date and Time attribute (17%), Patient ID (16%) and Device Identification (DID) (14%).

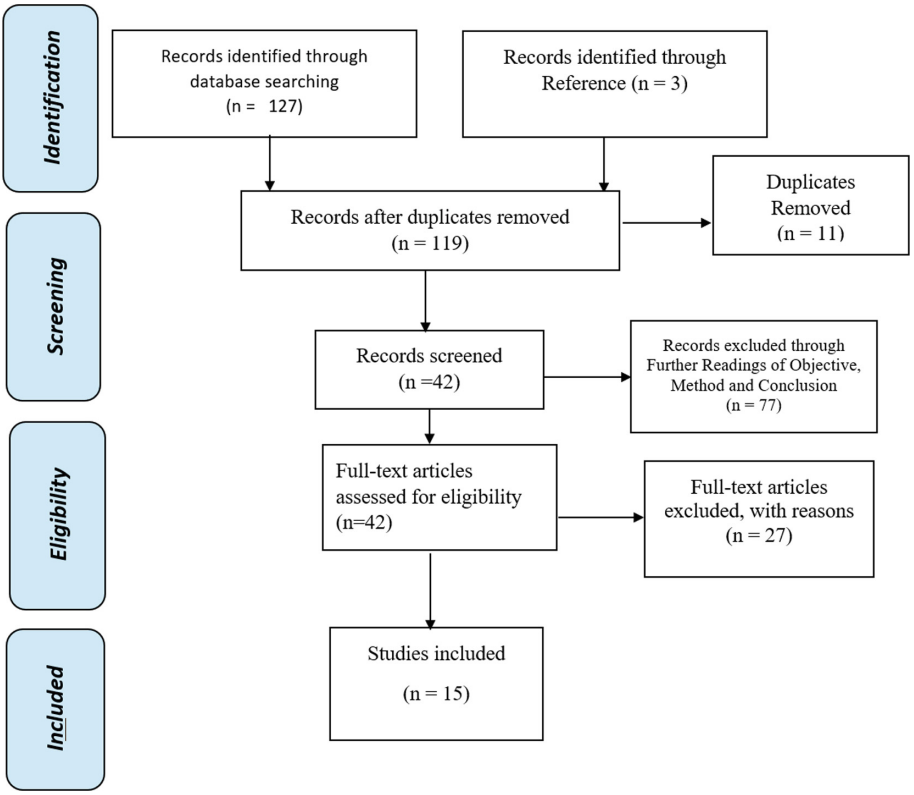


Fig. 1. Flowchart of the systematic review process

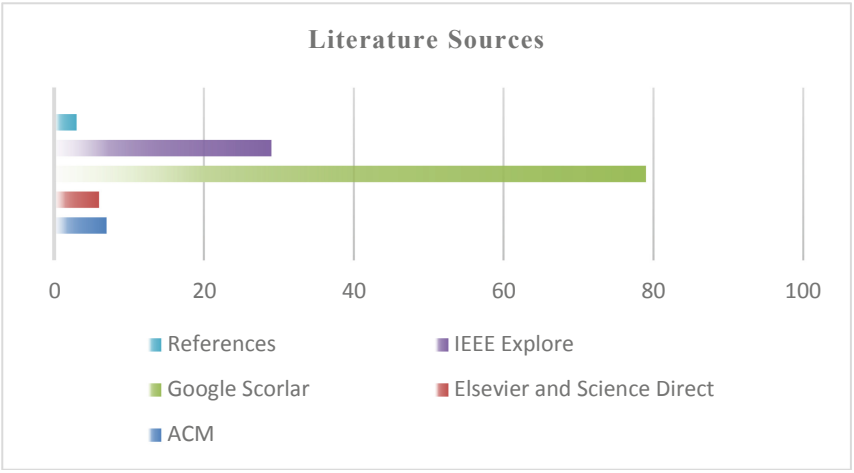


Fig. 2. Literature sources

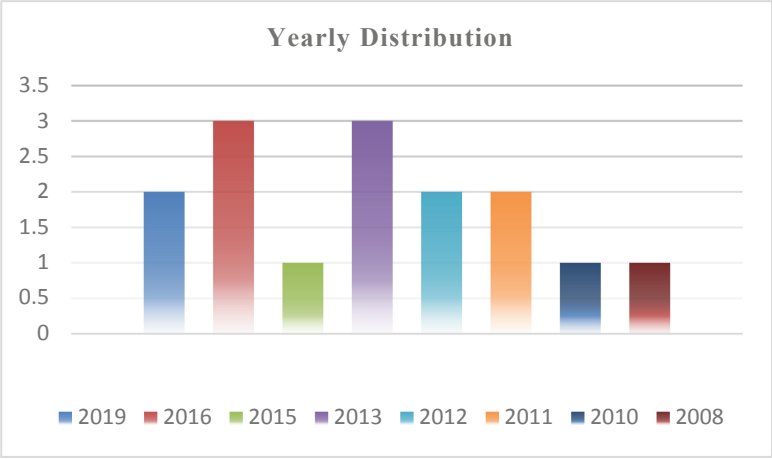


Fig. 3. Yearly distribution

Table 1. Algorithms, features, their related data sources and application domain

Study	Algorithms						Features							Data Sources				Application Domain	
	K-NN	Bayesian Network	Random Forest	J48	SVM	C4.5	User ID	Patient ID	Device ID	User Actions	Date and Time	Route	Location	EHR Logs	Host System Log	Network Logs	Key Stroke D.	Anomaly	Intrusion
[27]							■	■	■		■	■		■				■	
[28]							■			■					■				■
[29]							■	■			■			■					■
[30]							■			■									■
[31]							■	■	■		■	■	■	■				■	
[32]							■	■	■		■		■	■				■	
[33]							■	■	■		■					■			■
[34]							■	■	■	■	■					■			■
[35]							■		■							■		■	
[36]								■			■		■	■					■
[37]										■	■	■		■				■	
[38]							■	■	■	■	■			■				■	
[3]							■	■	■		■	■	■	■				■	
[39]																	■	■	
[40]							■	■	■		■			■				■	

Table 2. Algorithms and their respective proportions

Algorithm	Count	%
K-Nearest Neighbors (KNN) [27, 30, 31, 38, 40]	5	17
Bayesian Network (BN) [27, 30, 33, 39]	4	14
C4.5 [34, 37, 39]	3	10
Random Forest [34, 39]	2	7
J48 [37, 39]	2	7
Principal Component Analysis (PCA) [40]	2	7
Spectral project model [40]	1	3
SVM [39]	1	3
k-Means [28]	1	3
Spectral project method	1	3
Ensemble averaging and a human-in-the-loop model [35]	1	3
Partitioning around Medoids with k estimation (PAMK) [34]	1	3
Distance based model [32]	1	3
White-box anomaly detection system [29]	1	3
C5.0	1	3
Hidden Markov Model (HMM) [32]	1	3
Graph-Based [3]	1	3

Table 3. Features used

Feature	Count	%
User Identification (UID)	12	19
Patient Identification (PID)	10	16
Device ID(DID)	9	14
Access Control (AC)	5	8
Date and time	11	17
Location	4	6
Service/Route	5	8
Actions (Delete, Update, Insert, Copy, View)	3	5
Roles	3	5
Reasons	1	2

III. Data Sources

Most of the data sources were EHR logs (60%) and Network logs (20%) as shown in Table 4.

Table 4. Data sources used

Data source	Count	%
Electronic health records logs (EMR) logs	9	60
Host-based logs	1	7
Network logs	3	20
Key-stroke activities	1	7

IV. Performance Methods

Regarding performance methods as shown in Table 5, FP (23%), TP (20%) and Recall (13%) were mostly used to assess the studies.

Table 5. Performance methods

Performance methods	Count	%
True Positive (TP)	8	20
False Positive (FP)	9	23
False Negative (FN)	5	13
Receiver operating characteristic roc curve	5	13
Area Under ROC (AUC) curve	3	8
Recall (Sensitivity)	5	13
Precision	3	8
Accuracy	2	5

V. Application Scenario

The studies in the review were mostly applied for anomaly detection (60%) and Intrusion detection (40%) as shown in Fig. 4.

VI. Data Format

Regarding file format, Comma separated values (CSV) was commonly used as the file format [27, 28]. Some studies also used SQL file format [29, 41].

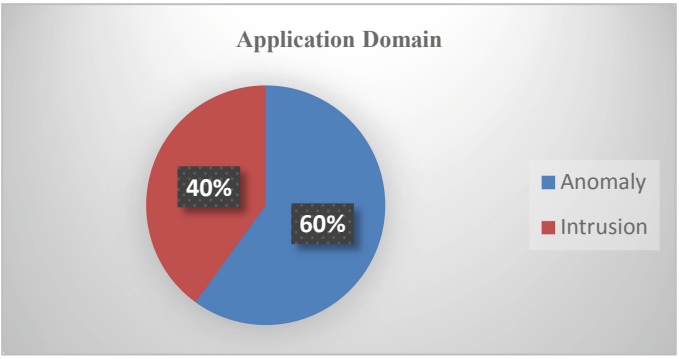


Fig. 4. Application domain

VII. Ground Truth

In the review, the ground truth was being established with similarity measures, observed and controlled practices and historical data of staffs' practices as shown in Table 6.

Table 6. Ground truth

Ground truth	Count	%
Similarity measures	3	38
Observed practices	3	38
Historical data	2	25

VIII. Privacy preserving data mining approach

Privacy preserving methods which were adopted in study are tokenization [27], deidentification [31] and removal of medical information [37].

IX. Nature of Data Source

With reference to Fig. 5, the nature of the data sources which were used in the studies were mostly Real data (80%) and synthetic data (20%).

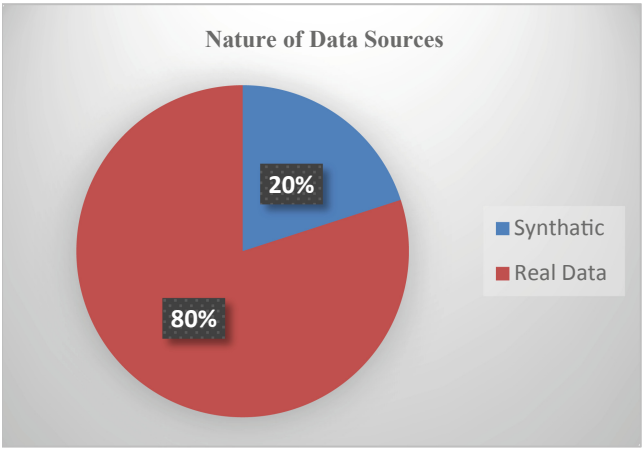


Fig. 5. Nature of data sources

5 Discussion

The main purpose of this systematic review was to find details of Artificial Intelligence (AI) methods and suitable healthcare staffs' generated security practice data that can be efficiently mined to determine the status of healthcare security practices with respect to required security practices. The main findings in the study are as shown in Table 7.

Table 7. Principal findings

Category	Most used
Algorithms	KNN and Bayesian networks
Features	User IDS, Patient IDs, Device ID, Date and time, Location, Route and actions
Data sources	Electronic health Records (EHR) logs and Network logs
Application domain	Anomaly detection
Performance methods	True Positive, False Positive, False Negative, ROC curve, AUC
Data format	CSV
Nature of data sources	Real data logs
Ground truth	Similarity measures and observed data
Privacy preserving approaches	Tokenization and deidentification

With reference to Fig. 1, 2 and 3 and Table 1, there were 15 studies which met the inclusion and exclusion criteria. Recently, a related systematic review for countermeasures against internal threats in healthcare also found about 5 machine learning methods,

[12] which were fit for such measures. This suggests that the adoption of AI methods for modeling and analyzing healthcare professionals' generated security practice data, is still an emerging topic of academic interest.

5.1 AI Methods

As shown in Table 2 and Table 7, various algorithms were identified in the study, but the most used methods were KNN and BN algorithms. K-Nearest Neighbors (kNN) is a supervised learning -based classification algorithm [30] which gets its intelligence from labeled data. The KNN then tries to classify unlabeled data item based on the category of the majority of most similar training data items known as K. The similarity between two data items in KNN, can be determined with the Euclidean distance of the various respective feature vectors of the data item. Another method which was mostly used is Bayesian Network (BN). BN is a probabilistic classifier algorithm, based on the assumption that related pair of features used for determining an outcome are independent of each other and equal [30]. There are two commonly used methods of BN for classifying text, thus the multi-variant Bernoulli and multinomial models. KNN and BN algorithms were mostly used based on their comparatively higher detection accuracy. For instance, in an experimental assessment of KNN and BNN for security countermeasures of internal threats in healthcare, both KNN and BN had over 90% accuracy. BN performed better (94%) than the KNN (93%). In a related study [12], the KNN method was found to have higher detection rate with high true positive rates and low false positive rate.

The major issue with KNN in the context of healthcare staff security generated data is the lack of appropriate labeled data [23, 35, 42]. Within the healthcare setting, emergencies often dictate needs. In such situations, broader accesses for resources are normally allowed, making it challenging for reliable labeled data [23, 35, 42]. Therefore, in adopting KNN for empirical studies, the availability of appropriate labeled data should be considered but, in the absence of labeled data, unsupervised clustering methods such as K means clustering could also be considered [26].

5.2 Input Data, Features, Sources, Ground Truth, Data Format and Nature of Data

The input data which was mostly used include EHR logs and Network data. A study which was conducted by Yeng et al., for observational measures towards profiling healthcare staffs' security practices, identified various sources including EHR logs, browser history, network logs, and patterns of keystroke dynamics [25]. Most EHR systems uses an emergency access control mechanism, known as "break the glass "or self-authorization" [43]. This enables healthcare staffs, to access patients' medical records during emergency situations without passing through conventional procedures for access authorization. A study into access control methods in Norway [43] revealed that about 50% of 100,000 patients records were accessed by 12,0000 healthcare staffs (representing about 45% of the users) through self-authorization. In such a scenario, EHR remains a vital source for analyzing for deviations of required healthcare security practices.

Regarding Ground Truth, it refers to the base-line, often used for training algorithms [44]. The detection efficiency of the algorithms can be negatively impacted if the accuracy

of the ground-truth is low. As shown in Table 6, various methods such as similarity measures, observed data and historical methods were used. Similarity measure compares security practices with other healthcare professionals who have similar security practices. Observed measure is a control approach of obtaining the ground truth whereby some users were observed to conduct their security practices under a supervised, required security practices [39]. But the historical data basically relied on past records with a trust that, the data is reliable enough for training set. These methods can be assessed for adoption in related studies.

EHR contains most of the features which were identified in this review as shown in Table 7. Features such as patients ID, Actions, and User ID are primary features in EHR logs. The actions of the users such as deletion, inserting, updating and various routes such as diagnosis, prescriptions, and drugs dispensing can be tracked in EHR logs [43].

5.3 Application Scenario and Privacy Preserving Log Analysis

The application of AI methods to analyze big data, generated by healthcare professional security practice, is a reactive approach. With such approaches, the primary aim is to determine deviations or outliers in healthcare security practices and further process these anomalies for possible malicious activities. As most of the algorithms were applied for anomaly detection (60%), such methods can be used to initially detect outliers. Deep leaning methods such as BN can then be used to further analyze the outliers for possible intrusions. This would help in privacy preserving at the same time while saving resources. Privacy preserving in data mining provides method to efficiently analyze data while shielding the identifications of the data subjects in a way to respect their right to privacy. For instance, limited number of less sensitive features can be used with KNN-based algorithms and if there exist outliers, BN methods can then be applied on only large number of the outliers to further assess these anomalies. In the review, deidentification, tokenization and sensitive data removals were some of the methods adopted to preserve privacy.

6 Conclusion

Based on the galloping rate of data breaches in healthcare, Healthcare Security Practice Analysis, Modeling and Incentivization (HSPAMI) project was initiated to observe, model and analyze healthcare staffs' security practices. One of the approaches in the project is the adoption of AI methods for modeling and analyzing healthcare staffs' generated security practice data. This systematic review was then conducted to identify, asses and analyze the appropriate AI methods and data sources. Out of about 130 articles which were initially identified in the context of human-generated healthcare data for security measures in healthcare, 15 articles were found to meet this inclusion and exclusion criteria. After the assessment and analysis, various methods such as KNN, Bayesian Network and Decision Trees (C4.5) algorithms were mostly applied on Electronic Health Records (EHR) Logs and Network logs with varying input features of healthcare staffs' security practices.

With these algorithms, security practice of healthcare staffs, can then be studied. Deviations of security practices from required healthcare staffs' security behavior can be examined to define appropriated incentives towards improving conscious care security practice. Analyzing healthcare staff security practice with AI seems to be a new research focus area and this resulted into the inclusion of only 15 articles in this study. Among these included articles, there were no adequate recorded performance scores. As a result, the study could not adequately perform a comparative assessment of the performance of the identified algorithms. Future work would include development of a framework and a practical assessment of the performance of these methods towards implementation in real healthcare staffs' generated logs.

References

1. Verison: Data breaches report (2019)
2. HealthITSecurity: The 10 Biggest Healthcare Data Breaches of 2019, So Far. @SecurityHIT (2019)
3. Zhang, H., Mehotra, S., Liebovitz, D., Gunter, C., Malin, B.: Mining deviations from patient care pathways via electronic medical record system audits. ACM Trans. Manage. Inf. Syst. (TMIS) **4**, 1–20 (2013)
4. Humer, C., Finkle, J.: Your medical record is worth more to hackers than your credit card (2014)
5. Humer, C., Finkle, J.: Your medical record is worth more to hackers than your credit card. Reuters, 24 September 2014
6. Connolly, L.Y., Lang, M., Gathegi, J., Tygar, D.J.: Organisational culture, procedural countermeasures, and employee security behaviour (2017). https://doi.org/10.1108/ICS-03-2017-0013
7. Tetz, E.: Network Firewalls: Perimeter Defense - dummies (2018)
8. Predd, J., Pfleeger, S.L., Hunker, J., Bulford, C.: Insiders behaving badly. IEEE J. Mag. **6**, 66–70 (2008)
9. Cannoy, S.D., Salam, A.F.: A framework for health care information assurance policy and compliance. Commun ACM. **53**(3), 126–131 (2010)
10. Safa, N.S., Sookhak, M., Von Solms, R., Furnell, S., Ghani, N.A., Herawan, T.: Information security conscious care behaviour formation in organizations. Comput. Secur. **53**, 65–78 (2015)
11. Yeng, P.K., Szekeres, A., Yang, B., Snekkenes, E.A.: Framework for healthcare staffs' information security practice analysis: psycho-socio-cultural context. J. Med. Internet Res. (2019)
12. Walker-Roberts, S., Hammoudeh, M., Dehghantanha, A.: A systematic review of the availability and efficacy of countermeasures to internal threats in healthcare critical infrastructure. IEEE Access. **6**, 25167–25177 (2018)
13. Böse, B., Avasarala, B., Tirthapura, S., Chung, Y., Steiner, D.: Detecting insider threats using RADISH: a system for real-time anomaly detection in heterogeneous data streams. IEEE Syst. J. **11**(2), 471–482 (2017)
14. Gafny, M., Shabtai, A., Rokach, L., Elovici, Y.: Detecting data misuse by applying context-based data linkage, pp. 3–12 (2010)
15. Chen, Y., Nyemba, S., Zhang, W., Malin, B.: Specializing network analysis to detect anomalous insider actions. Secur. Inf. **1**(1), 1–24 (2012)

16. Islam, S., Hasan, M., Wang, X., Germack, H.D., Noor-E-Alam, M.: A systematic review on healthcare analytics: application and theoretical perspective of data mining. Healthcare (Basel) **6**(2), 54 (2018)
17. Gheyas, I., Abdallah, A.: Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis. Big Data Analytics **1**, 6 (2016)
18. Ghafir, I., Husák, M., Prenosil, V.: A survey on intrusion detection and prevention systems (2014)
19. Shaban-Nejad, A., Michalowski, M., Buckeridge, D.: Health intelligence: how artificial intelligence transforms population and personalized health. Nat. Med. **50** (2018)
20. Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., et al.: Artificial intelligence in healthcare: past, present and future. BMJ (Clinical research ed), 2 (2017). svn-2017
21. Wahl, B., Cossy-Gantner, A., Germann, S., Schwalbe, N.: Artificial intelligence (AI) and global health: How can AI contribute to health in resource-poor settings? BMJ Global Health **3**, e000798 (2018)
22. Vihinen, M., Samarghitean, C.: Medical expert systems. Curr. Bioinf. **3**(1), 56–65 (2008)
23. Chandra, S., Ray, S., Goswami, R.T.: Big data security in healthcare: survey on frameworks and algorithms, pp. 89–94 (2017)
24. Code of conduct for information security and data protection in the healthcare and care services sector (2018)
25. Yeng, P., Yang, B., Snekkenes, E. (eds.): Observational measures for effective profiling of healthcare staffs' security practices. In: IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 15–19 July 2019
26. PRISMA: PRISMA 2018. http://www.prisma-statement.org/
27. Boddy, A.J., Hurst, W., Mackay, M., Rhalibi, A.: Density-based outlier detection for safeguarding electronic patient record systems. IEEE Access **7**, 40285–40294 (2019)
28. Tchakoucht, T.A., Ezziyyani, M., Jbilou, M., Salaun, M., (eds.): Behavioral approach for intrusion detection. In: IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), 17–20 November 2015
29. Costante, E., Fauri, D., Etalle, S., Hartog, J.D., Zannone, N., (eds.): A hybrid framework for data loss prevention and detection. In: IEEE Security and Privacy Workshops (SPW), 22–26 May 2016
30. García Adeva, J.J., Pikatza Atxa, J.M.: Intrusion detection in web applications using text mining. Eng. Appl. Artif. Intell. **20**(4), 555–566 (2007)
31. Gupta, S., Hanson, C., Gunter, C.A., Frank, M., Liebovitz, D., Malin, B., (eds.): Modeling and detecting anomalous topic access. In: IEEE International Conference on Intelligence and Security Informatics, 4–7 June 2013
32. Li, X., Xue, Y., Malin, B., (eds.): Detecting anomalous user behaviors in workflow-driven web applications. In: IEEE 31st Symposium on Reliable Distributed Systems, 8–11 October 2012
33. Amálio, N., Spanoudakis, G., (eds.): From monitoring templates to security monitoring and threat detection. In: Second International Conference on Emerging Security Information, Systems and Technologies, 25–31 August 2008
34. Pierrot, D., Harbi, N., Darmont, J., (eds.): Hybrid intrusion detection in information systems. In: International Conference on Information Science and Security (ICISS), 19–22 December 2016
35. Boddy, A., Hurst, W., Mackay, M., Rhalibi, A.E., (eds.): A hybrid density-based outlier detection model for privacy in electronic patient record system. In: 5th International Conference on Information Management (ICIM), 24–27 March 2019
36. Asfaw, B., Bekele, D., Eshete, B., Villafiorita, A., Weldemariam K., (eds.): Host-based anomaly detection for pervasive medical systems. In: 2010 Fifth International Conference on Risks and Security of Internet and Systems (CRiSIS), 10–13 October 2010

37. Ziemniak, T., (ed.): Use of machine learning classification techniques to detect atypical behavior in medical applications. In: Sixth International Conference on IT Security Incident Management and IT Forensics, 10–12 May 2011
38. Chen, Y., Nyemba, S., Malin, B.: Detecting anomalous insiders in collaborative information systems. IEEE Trans. Dependable Secure Comput. **9**, 332–344 (2012)
39. Wesołowski, T., Porwik, P., Doroz, R.: Electronic health record security based on ensemble classification of keystroke dynamics. Appl. Artif. Intell. **30**, 521–540 (2016)
40. Chen, Y., Malin, B.: Detection of anomalous insiders in collaborative environments via relational analysis of access logs, pp. 63–74 (2011)
41. Asfaw, B., Bekele, D., Eshete, B., Villafiorita, A., Weldemariam, K.: Host-based anomaly detection for pervasive medical systems pp. 1–8 (2010)
42. Gates, C., Li, N., Xu, Z., Chari, S., Molloy, I., Park Y. Detecting insider information theft using features from file access logs, pp. 383–400 (2014)
43. Røstad, L., Edsberg, O.: A study of access control requirements for healthcare systems based on audit trails from access logs, pp. 175–186 (2006)
44. Smyth, P., Fayyad, U., Burl, M., Perona, P., Baldi, P.: Inferring ground truth from subjective labelling of venus images. In: Advances in Neural Information Processing Systems, p. 7 (1996)

Detection of Anomalous Patterns in Water Consumption: An Overview of Approaches

José Carlos Carrasco-Jiménez[1(✉)], Filippo Baldaro[2(✉)], and Fernando Cucchietti[1(✉)]

[1] Barcelona Supercomputing Center, Barcelona, Spain
{jose.carrasco,fernando.cucchietti}@bsc.es
[2] CETAQUA Water Technology Center, Barcelona, Spain
fabaldaro@cetaqua.com
http://ris3catutilities.com/eng/index_ris3cat.html

Abstract. The water distribution system constantly aims at improving and efficiently distributing water to the city. Thus, understanding the nature of irregularities that may interrupt or exacerbate the service is at the core of their business model. The detection of technical and non-technical losses allows water companies to improve the sustainability and affordability of the service. Anomaly detection in water consumption is at present a challenging task. Manual inspection of data is tedious and requires a large workforce. Fortunately, the sector may benefit from automatized and intelligent workflows to reduce the amount of time required to identify abnormal water consumption. The aim of this research work is to develop a methodology to detect anomalies and irregular patterns of water consumption. We propose the use of algorithms of different nature that approach the problem of anomaly detection from different perspectives that go from searching deviations from typical behavior to identification of anomalous pattern changes in prolonged periods of time. The experiments reveal that different approaches to the problem of anomaly detection provide complementary clues to contextualize household water consumption. In addition, all the information extracted from each approach can be used in conjunction to provide insights for decision-making.

Keywords: Anomaly detection · Water consumption · Time series · Decision making

1 Introduction

Late payment of bills as well as non-technical losses (NTL) derived from commercial fraud has increased in most European countries, leading to a critical point in the financial management of operators in the utilities sector.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 19–33, 2021.
https://doi.org/10.1007/978-3-030-55180-3_2

NTL in the utilities sector leads to the necessity of improving the efficiency of the process of detecting anomalous consumption and the location of non-technical losses. Identifying the origin of such losses allows companies to offer new payment solutions that are customized to the needs of the customers, guaranteeing the economic sustainability of the service as well as the affordability for those with economic difficulties.

Anomaly detection in water consumption is at present a challenging task. Data is generally dispersed and estimated readings and manual readings taken bimonthly or quarterly, usually result in difficult scenarios in which the task of classifying the type of anomaly is hard to predict. For the water company understanding the nature of the anomaly is of avid importance. For example, if an anomalous consumption is due to a broken pipe or a leakage, the result of classifying the anomaly as a technical loss leads to the quick repair of the equipment. On the other hand, if the anomaly is due to an intended manipulation at the water meter, a corrective action can be taken.

The deployment of smart-meters and the development of algorithms to process the data generated by these devices opens up new opportunities in the rapid identification of anomalous consumptions. This allows the operator to take action through the application of search criteria and the selection of possible cases of water pipe breakdown or irregular consumption, leading to savings in the cost of operations and an optimal, as well as, an efficient supply of water. As a consequence, the number of irregular consumptions is expected to decrease.

The manual identification of anomalous consumptions is conventionally done as follows:

1. identify water bills that have values out of the expected range given the region and the type of customer (e.g. industrial, residential);
2. prioritize the verification given a number of rules;
3. verify the service points that have been marked as possible anomalies due to identification of anomalous readings;
4. take a sample of the bills of a given user that is suspected of presenting anomalous consumption;
5. request a visual inspection of those pieces of equipment that are likely to show anomalous behavior (either due to technical or non-technical issues).

The process of determining anomalous behaviors in water consumption can be a tedious business given the large amount of data that needs to be processed and the large volume of clients. The cost of manually identifying possible anomalies can be diminished if the workflow can be automatized and supplied with intelligent algorithms. The development of new techniques to detect anomalous behaviors in water consumption has a direct impact in the business model of water companies. Among the benefits we can find targeted inspections with higher probabilities of identifying technical or non-technical issues, identification of metrological inefficiencies (e.g. manipulated meters), and identification of misclassified clients (i.e. incorrect type of tariff).

The aim of this research work is to develop a methodology to detect anomalies and irregular behavior from water consumption time series. This work is limited to the clues that can be exploited from a time series, considering relevant domain knowledge such as the types of clues that indicate that a consumption is potentially out of the normal behavior including point anomalies and pattern anomalies found in the time series.

The remainder of this paper is organized as follows. Section 2 describes the background information and Sect. 3 describes the related works. Sect. 4 details the methodology, including a description of the dataset and the data processing workflow. Sect. 5 describes the fundamental model in which manual inspection is based. Furthermore, Sect. 6 elaborates on the analysis of point anomalies found in time series of water consumption while Sect. 7 presents the result of anomalous behavior detection algorithms. We conclude with a discussion of the most notable findings and future work in Sect. 9.

2 Background

In this section we describe the background information required to contextualize the problem at hand. We start by defining in a concise manner how the water distribution network works. An explanation of definition and possible cause of anomalies in household water consumption.

2.1 Water Balance of a Drinking Water Distribution System

In the water sector, Unregistered Water (UW) indicates water that has been *produced*, i.e. processed in a water treatment plant and supplied into the distribution network, and is lost before reaching the end clients. Furthermore, Registered Water (RW) or Billed Water (BW) indicates the amount of water supplied and registered by the automated meter reading devices deployed at the residential level or municipal distribution point.

The International Water Association (IWA) proposes categories to evaluate the different components of UW [1]:

- real or physical losses corresponding mainly to leakage in the transportation network, the reservoir, the distribution network, or end users
- apparent or commercial losses including Authorized Consumption Unregistered (ACU), fraud or erroneous readings

The ACU consumptions correspond to the water used in the operations of network maintenance, or agreements with the public sector including firefighters or park irrigation. Figure 1 summarizes the water balance of a drinking water distribution system from the utilities perspective. The dotted-circle represents the part of the distribution system where technical and non-technical losses are to be inferred.

Frauds are defined as irregular connections to the distribution network, non-regulated discharge, or manipulation of smart-meters with the aim of benefiting

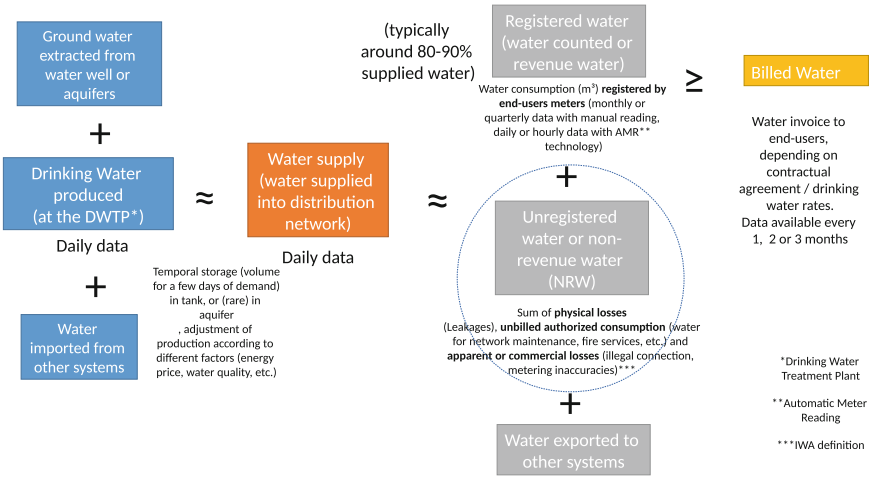


Fig. 1. Water balance of a drinking water distribution system (simplified, water utilities perspective).

with lower charges. Moreover, a distinct usage from that contracted fare, usually applying a lower fare than that required.

Lastly, lack of precision in the metering devices can be due to poor installations, underestimated volumes of water leading to smaller devices than needed, leakage, or aged devices. We can quantify the losses due to fraud or underestimation through the mean monetary value of the water service in Spain. The mean price in Spain is 2,24 €/m^3 from which €1,23 correspond to the supply service, €0,32 stand for sewage system, and €0,69 for depuration. Using these reference values we can estimate the commercial loss of the operators due to fraud or underestimation, leading to a value of approximately 730 million €/year.

2.2 Anomalies

In [2], anomalies are defined as patterns that do not conform to the normal expected behavior. As a consequence, we need to identify which consumptions are to be defined as normal.

The anomalies associated to the consumption of water can be categorized as follows:

1. data transmission errors,
2. change in the behavior of water consumption,
3. fraud or manipulation of the smart-meters.

Furthermore, associating a given pattern to a deviation from the expected behavior of water consumption remains a challenging task. In other words, defining what the normal behavior looks like depend much on the context, not only of a single user but of the patterns exhibited in the region at hand.

[2] defines a few challenges that raise from the sheer definition of anomaly. The first thing the authors identify as challenging is defining what is the normal expected behavior. In the case of water consumption, normal behaviors tend to evolve and expose a cyclical tendency leading to different definitions of *normality*, understood as more frequent. In this research work, we deal with time series that exhibit patterns that may be suggested as deviation from the normal (i.e. more frequent) consumption when considered outside of its context. Nevertheless, when the patterns are analyzed inside their context, those punctual patterns can be categorized as normal behavior. In other words, anomalous behavior is determined within the limits of a specific context (i.e. region and fare).

3 Related Work

The notion of anomaly has several applications in the water sector. Given the nature of the data, different challenges arise and a variety of anomaly detection techniques apply in different scenarios. Anomaly detection serves different purposes that go from general water analytics to more specific applications such as water quality, identification of changes in household consumption, and detection of technical and non-technical losses in the water distribution network.

In [3], the authors propose a constrained-clustering-based approach for anomaly detection in real-world monitoring time series, a task that cannot be manually executed or analyzed given the large number of customers managed, even by small operators. Similar applications have been reported for hydrological time series in which abnormal patterns play an important role in the analysis and decision making [4].

With regard to water quality, operators are interested in understanding deviations from what they consider *normal* values of quality. The search for deviations is considered a problem of anomaly detection. Identifying anomalous points in drinking-water quality data may lead to improved quality. Recent works [5,6] seek to improve water quality by identifying anomalies that need to be eliminated or avoided.

Determining anomalous behaviors in water consumption is also a major task in water analytics. For example, in [7], changes in household water consumption are identified from water usage readings with the aim of detecting anomalies that may lead to the discovery of manipulated meters. Similarly, [8,9] propose two different approaches for determining anomalous behaviors in water consumption: punctual anomalous consumption and anomalous changes in behavior of water consumption, respectively.

Other interesting challenges in which the detection of anomalies may shed some light is the identification of technical and non-technical losses, which is of avid concern for water operators. For instance, [10] describes the implementation of a methodology for the automated detection of pipe bursts and other events that induce similar abnormal pressure/flow variations (technical losses). Similarly, in [11], authors use machine learning to identify malfunctioning meters and leaks (considered technical losses), as well as fraud (considered non-technical

loss). In this respect, [12] proposes the use of an anomaly detection approach based on unsupervised clustering algorithms employed to identify non-technical losses. A more recent work combines anomaly detection techniques with computer vision strategies to detect fraud in water meters, such is the case of [13], in which water meter seals are analyzed through morphological image processing. Then pattern recognition techniques serve to identify suspected irregularities and anomalies in water meters.

4 Methods

In this section we describe the methodology proposed to reach the objectives of this research work. We start by describing the dataset, followed by a description of the data processing.

4.1 Dataset Description

The main source of information for the detection of anomalous behavior in water consumption comes from two types of databases: 1) *Customer Information System (CIS)* and 2) *Customer Relationship Management (CRM)*, both are commercial management systems.

The smart-meters, also known as *Automated Meter Reading (AMR)*, collect and transmit data at different rates, although data is transmitted generally on a daily basis.

The dataset comprises over 1.460.000 records, totaling more than 4000 customers from two different regions over a period of one year. Customers are assigned a tariff based on the average volume of water consumed. Different regions may contain different types of fares depending on the kind of profiles present in each region. Part of the analysis of this work will make use of the region and type of tariff assigned to each client, in order to understand the customer's behavior with respect to the typical consumption behavior of the region and tariff.

For this work, data has been previously anonymized and pre-processed by the water company. Each record collected contains three pieces of information as summarized in Table 1.

Table 1. Summary of data collected for each recording.

Data	Description	Example
CUSTOMER	Anonymized customer ID	AC11UA487082
DATE	Date in the format *yyyy-mm-dd*	2014-01-11
VOLUME	Amount of water consumed (liters)	176

Given the data structure described in Table 1, the sequence of consumption can be reconstructed to show the evolution through time. The construction of the time series is the foundation to discover anomalies.

4.2 Data Processing

The readings transmitted from the smart-meters follow three sequential phases:

1. the transmitter reads the datum from the automated meter reading every hour (or every six hours depending on the configuration of the AMR),
2. the data is sent to the CIS on a daily basis in chunks of 4 or 24 data per transmitter,
3. the data is centralized in the Master Data Management (MDM).

The data collected from the smart-meters are indices of consumption that are later transformed into consumption by the difference of two indices. In other words, the consumption is computed as the difference between the index registered at the beginning and the index registered at the end of the reading cycle.

The following step after the transformation from indices into consumption is the construction of time series, which are later fed into the algorithms developed in this research work.

The anomaly detection algorithms depend on the context to which the labels are available. In Sect. 1 we defined the steps executed in a manual inspection of potential customers who present anomalous behaviors in water consumption. As a consequence some customers have been manually analyzed but the number of labeled samples remains insufficient to use supervised anomaly detection.

On the other hand, unsupervised anomaly detection algorithms are widely applicable to situations where we lack labeled instances. Unsupervised techniques assume that those instances that are more frequent in a dataset can be considered as normal instances, leading to the identification of instances that deviate from normality, also called anomalies.

5 Deviation from Expected Normal Water Consumption Levels

In Sect. 4.2, we described how the data is transformed from its raw format into time series of water consumption. The sequence of consumption points is the input of the algorithms analyzed in this work.

An anomaly can be defined as a pattern that does not conform to the expected normal behavior [2]. In the scenario at hand, an anomaly is a deviation from the expected normal water consumption levels. A first approach to understanding anomalous levels of water consumption is to consider the time series of water consumption of a client and analyze it with respect to the context of the customer in accordance with the definitions in Sect. 2.2.

In other words, a segmentation of the dataset based on similar environmental and social characteristics, as well as similar tariffs, is needed. In our first approach we characterize the detection of deviation from expected normal water consumption levels as described in Algorithm 1.

The pseudocode described starts by splitting the dataset by region and tariff. The next step requires the estimation of the mean consumption of each group

Algorithm 1: Procedure to detect deviations from expected normal water consumption levels.

Input: threshold=3
Output: anomalies
Data: waterConsumptionData

```
1 groups ← splitByRegionAndFare(Data)
2 for group ← groups do
3     groupMean ← mean(group)
4     groupSd ← sd(group)
5     for c ← group do
6         upperThreshold ← groupMean + threshold * groupSd
7         lowerThreshold ← groupMean - threshold * groupSd
8         anomalies ← which( c ≥ upperThreshold || c ≤ lowerThreshold )
```

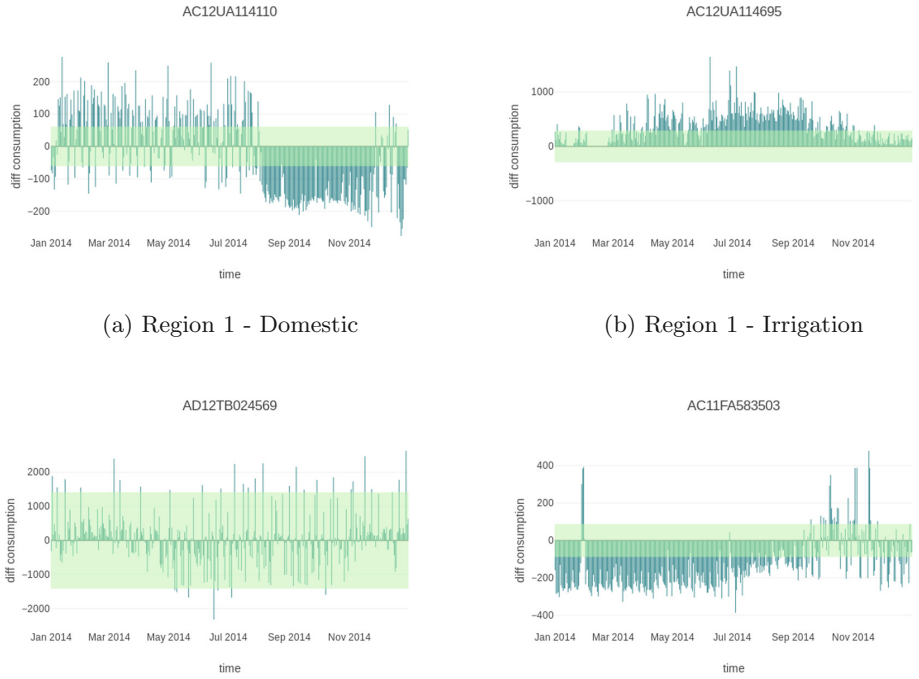
of customers as well as the standard deviation of each group. Assuming we have established a threshold to discriminate typical from anomalous values, which is commonly three standard deviations from the mean to be considered anomalous, we proceed to identify which points lie above the upper threshold and below the lower threshold. The aforementioned algorithm was applied to every customer in the database used for this research work. The resulting anomalies for a sample of customers is shown in Fig. 2.

For instance, in Fig. 2, the vertical bars show the deviations from the mean water consumption for the region and tariff to which the customer belongs. The vertical bars that lie above the zero line are the instances of time in which the water consumption is above the mean consumption of the region and those that lie below are the instances in which the consumption is below the mean consumption.

The light green rectangle drawn on each of the plots are interpreted as the maximum and minimum permitted deviations from the expected normal water consumption levels. In other words, vertical bars that lie outside the light green rectangle are to be considered as anomalous consumptions.

Consider the customer shown in Fig. 2 (c) which belongs to *region 2* and is assigned a tariff of the type *city*. As it is observed, most of the instances of time, the customer showed what would be classified as normal behavior of water consumption. On the other hand, the rest of customer in the sample exhibit many instances in which the consumption was anomalous.

In the next sections we will propose other complementary methods to detect anomalies. The information that can be extracted from this type of anomaly detection algorithm will serve to feed, and complement other sources of information, a meta-model proposed as future work.



(a) Region 1 - Domestic

(b) Region 1 - Irrigation

(c) Region 2 - City

(d) Region 2 - Industrial

Fig. 2. Difference from mean consumption. It shows the comparison between the consumption of a customer with respect to the mean consumption of the region and fare to which the customer belongs. A sample of customers from different regions and fares are shown.

6 Punctual Anomalous Consumption

Punctual anomalies refer to those deviations that happen at a given timestamp regardless of the prior and posterior values. This type of anomaly refers to values that are either too high or too low with respect to the rest of the time series, and occur at single points in time.

In this part of the work we analyzed three different methods for anomaly detection in a time series, two of which are based on STL decomposition (mean and median) [14] and one based on the Seasonal Hybrid ESD Test [15].

The three techniques provide different number of anomalies. Figure 3 shows the Mean-STL Decomposition, Median-STL Decomposition and Seasonal Hybrid ESD Test, from top to bottom respectively, of customer identified as AC11UA487082. The results show that Median-STL Decomposition approach identifies too many points as anomalous, a tendency shown by many customers, while Seasonal Hybrid ESD Test exhibits a more moderate solution as can be visually inspected in the aforementioned figure.

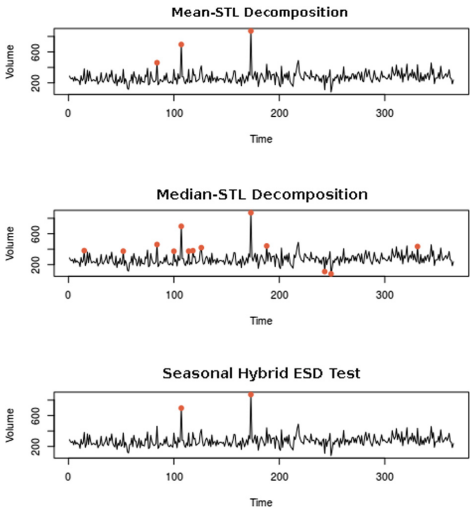


Fig. 3. Three different methods for point anomaly detection have been tested. Different techniques identify different number of anomalies for customer AC11UA487082.

On the other hand, in Fig. 4, Seasonal Hybrid ESD Test shows the worst viable solution given the large amount of anomalous points detected. Note that the Mean-STL Decomposition remains the most trustworthy.

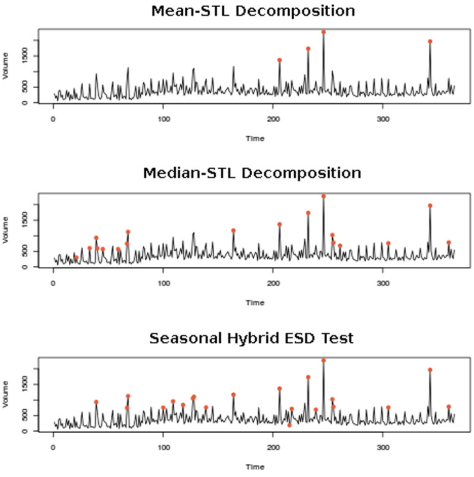


Fig. 4. Three different methods for point anomaly detection have been tested. Different techniques identify different number of anomalies for customer AC12UA114190.

In order to provide a better approximation to the most viable anomaly detection technique, an evaluation on all the customers was performed. Figure 5 shows the performance of each algorithm in terms of the number of anomalies detected

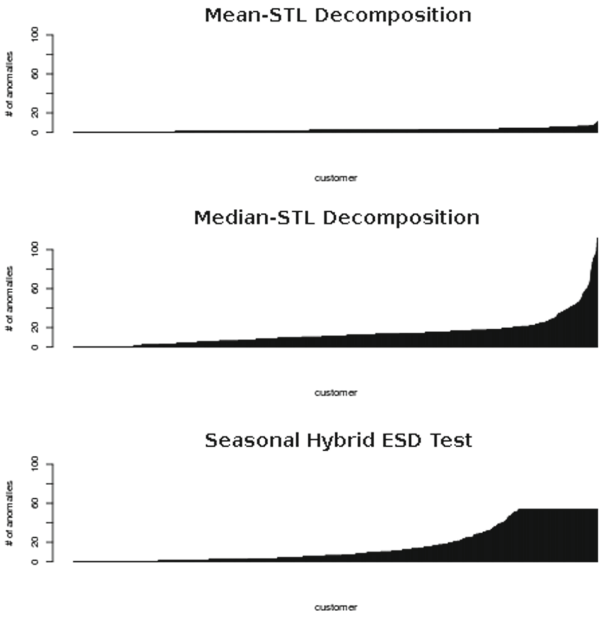


Fig. 5. The three different methods studied exhibit different sensitivity. Mean-STL Decomposition is more robust to small changes as defined by the smaller number of anomalies detected.

for each customer. If we look carefully at the distribution of anomalies we note that Mean-STL Decomposition has a lower number of anomalies in general.

Although, in general, Mean-STL Decomposition shows a more feasible solution, it would be interesting to know in which situations each technique is more viable.

7 Anomalous Pattern Changes in Household Water Consumption

As it was described in Sect. 3, changes in household water consumption are of avid importance for water operators. Point anomaly detection provides information about single instances of time in which an abnormal value is identified. But that is not the whole story. Another approach to anomaly detection is the identification of portentous changes in patterns of household water consumption. That is, changes in the patterns of consumption as opposed to single point anomalies. This approach is rather concerned with significant changes in prolonged periods of time, avoiding the raise of alarms when a single point has gone out of range. Although the approaches described in Sects. 5 and 6 are of different nature, the three approaches described in this paper do not compete against each other,

rather they can be used as complementary methods that perform on different scenarios or provide information about anomalies from different perspectives.

By understanding how medium to long-term behavior of water consumption, some non-technical losses can be detected. Take for example the case of an individual whose mean consumption drops drastically and remains stable for prolonged periods of time. In that case, point anomalies could not have detected a possible fraud. On the other hand, significant changes in the tendency may also indicate a technical loss when the mean consumption increases significantly.

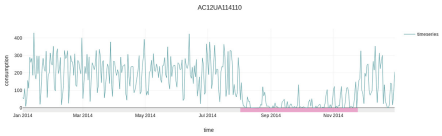
In order to detect anomalous changes in household water consumption, we propose the use of algorithms for the detection of multiple change points in multivariate time series. Change point detection can be seen as a model selection problem that consists of three elements, as defined in [16]:

1. cost function (e.g. maximum likelihood estimation, kernel-based detection),
2. search method (optimal, approximate),
3. constrains (i.e. penalty functions).

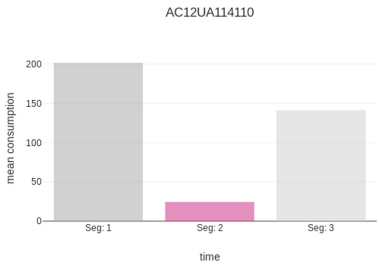
The most crucial part is the selection of search method. The method depends largely on two pieces of information: precision desired and whether or not the number of change points are known. With this in mind, we selected the Pruned Exact Linear Time (PELT) algorithm [17] which works for unknown number of change points and has better performance than the naive approach which proves to be computational cumbersome because of the quadratic complexity. Furthermore, the cost function is related to the type of change to detect. In our case, we used a non-parametric cost function, namely Mahalanobis metric.

Figure 6 shows the resulting segments extracted after applying the change point detection algorithm. The left side of the figure, i.e. Fig. 6 (a) and (c) display the temporal evolution of water consumption for two customer from different regions and types of tariffs. The bottom bars with alternating colors (gray and red) show the segments with different behaviors.

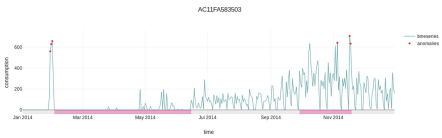
The corresponding figures on the right hand side of Fig. 6 show the mean consumption of a given customer at each segment. The order of the bars correspond to the order of the bottom bars of the left hand side of the figure. As we can see, the segments exhibit different behaviors, showing changes in household water consumption. To measure the significance of the changes observed, we need to consider two pieces of information: 1) mean consumption difference among the segments detected by the algorithm and 2) length of each segment. As part of the future work we will measure statistical significance from one segment to the other. This will allow us to identify possible cases with higher accuracy. Future work will also include a study to measure the minimum length of each segment to be defined as independent from the neighboring segments.



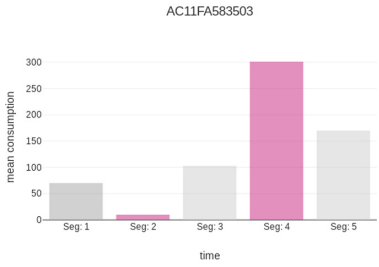
(a) Region 1 - Domestic - Consumption



(b) Region 1 - Domestic - Segments



(c) Region 2 - Industrial - Consumption



(d) Region 1 - Industrial - Segments

Fig. 6. Water consumption of customers. The red dots indicate days in which anomalous consumptions were identified.

8 Discussion and Limitations

The algorithms presented in this work are of different nature and have distinct capabilities. Each method approaches the contextualization of water consumption from different perspectives, emphasizing diverse aspects of the consumption. Although distinct, they offer complementary clues to contextualize household water consumption, leading to two broad categories of anomaly definitions, namely, those that are considered anomalous relative to the patterns of the region and tariff, and those that are tagged as anomalous when compared against their own consumption patterns. Those anomalies that are defined as such when a consumption exhibits deviations from expected normal water consumption levels of a given region and tariff fall in the first category. In this case, the context is defined by the behavior of water consumption of other customers. On the other hand, in punctual anomalies and anomalous pattern changes in household consumption, the context of the anomalies is dictated by past and future consumption of the same user, and does not depend on the behavior of others.

As it is explained in Sect. 9, the different approaches provide complementary clues to contextualize household water consumption and serve as the basis for a framework that is currently being designed. In fact, the methodology of this work sets the foundation of a broader goal, which consists in identifying households suspected of anomalous water use either due to technical or non-technical issues.

The scope of this work is limited to finding anomalous consumptions. It does not aim at interpreting the cause of each anomaly. Furthermore, each method provides different clues that may suggest whether the household presents anomalous patterns of water consumption due to technical issues or non-technical flaws, but identifying specific causes will be studied as part of the future work.

9 Conclusion and Future Work

Technical and non-technical problems derive in the loss of large quantities of resources that have a direct impact on the water supply chain. As it was stated in Sect. 2.2 the commercial loss can be estimated as much as 730 million €/year. Avoiding technical and non-technical losses remains a challenging tasks. Manual inspection of individual cases is a tedious task due to the large volumes of data. As a consequence, the water sector can benefit from an automatized workflow leading to targeted inspections with higher probabilities of identifying technical or non-technical issues.

In this work we developed a methodology to detect anomalous household water consumption and irregular patterns of consumption that may fall into fraudulent activities. The methodology consists of three approaches of different nature and capabilities. In the first part we implemented an algorithm to detect deviations from typical behavior. Although this piece of information can be useful, different perspectives can provide a richer set of conclusions. The second approach is defined as point anomaly detection in which punctual anomalies are detected regardless of the prolongation in time. Furthermore, to complement the previous approaches, a method to identify anomalous pattern changes in household water consumption was considered.

The different approaches provide complementary clues to contextualize household water consumption and serve as the basis for a framework that is currently being designed. Future work includes the development of a meta-model capable of classifying the type of anomaly, thus identifying whether the anomaly is due to a technical flaw or a non-technical issue. Other data sources will be incorporated including trend and seasonality of the demand of water in the region, weather information, holidays, reports of manual inspection of smart-meters, the state of the water distribution network, and authorized non-registered consumption (such as park irrigation).

Acknowledgment. This research work is cofounded by the European Regional Development Fund (FEDER) under the FEDER Catalonia Operative Programme 2014–2020 as part of the R+D Project from RIS3CAT Utilities 4.0 Community with reference code COMRDI16-1-0057.

References

1. Lambert, A., Hirner, W.: Losses from water supply systems: standard terminology and recommended performance measures, Report P-13, International Water Association (2000)

2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**, 1–58 (2009)
3. Vercruyssen, V., Meert, W., Verbruggen, G., Maes, K., Baumer, R., Davis, J.: Semi-supervised anomaly detection with an application to water analytics. In: IEEE International Conference on Data Mining (ICDM), pp. 527–536, November 2018
4. Sun, J., Lou, Y., Ye, F.: Research on anomaly pattern detection in hydrological time series. In: 14th Web Information Systems and Applications Conference (WISA), pp. 38–43, November 2017
5. Dogo, E.M., Nwulu, N.I., Twala, B., Aigbavboa, C.: A survey of machine learning methods applied to anomaly detection on drinking-water quality data. Urban Water J. **16**(3), 235–248 (2019)
6. Muharemi, F., Logofătu, D., Leon, F.: Machine learning approaches for anomaly detection of water quality on a real-world data set. J. Inf. Telecommun. **3**(3), 294–307 (2019)
7. Quinn, S., Murphy, N., Smeaton, A.F.: Tracking human behavioural consistency by analysing periodicity of household water consumption. In: Proceedings of the 2019 2nd International Conference on Sensors, Signal and Image Processing, SSIP 2019, New York, NY, USA, pp. 1–5. Association for Computing Machinery (2019)
8. González-Vidal, A., Cuenca-Jara, J., Skarmeta, A.F.: IoT for water management: towards intelligent anomaly detection. In: IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 858–863, April 2019
9. Christodoulou, S.E., Kourti, E., Agathokleous, A.: Waterloss detection in water distribution networks using wavelet change-point detection. Water Resources Manage. **31**, 979–994 (2017)
10. Romano, M., Kapelan, Z., Savić, D.A.: Automated detection of pipe bursts and other events in water distribution systems. J. Water Resour. Plann. Manage. **140**(4), 457–467 (2014)
11. Kermany, E., Mazzawi, H., Baras, D., Naveh, Y., Michaelis, H.: Analysis of advanced meter infrastructure data of water consumption in apartment buildings. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, New York, NY, USA, pp. 1159–1167. Association for Computing Machinery (2013)
12. Júnior, L.A.P., Ramos, C.C.O., Rodrigues, D., Pereira, D.R., de Souza, A.N., da Costa, K.A.P., Papa, J.P.: Unsupervised non-technical losses identification through optimum-path forest. Electr. Power Syst. Res. **140**, 413–423 (2016)
13. Detroz, J.P., da Silva, A.T.: Fraud detection in water meters using pattern recognition techniques. In: Proceedings of the Symposium on Applied Computing, SAC 2017, New York, NY, USA, pp. 77–82. Association for Computing Machinery (2017)
14. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: a seasonal-trend decomposition procedure based on loess. J. Off. Stat. **6**, 3–37 (1990)
15. Vallis, O., Hochenbaum, J., Kejariwal, A.: A novel technique for long-term anomaly detection in the cloud. In: Proceedings of the 6th USENIX Conference on Hot Topics in Cloud Computing, HotCloud 2014, USA, p. 15. USENIX Association (2014)
16. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Sig. Process. **167**, 107299 (2020)
17. Wambui, G.D., Waititu, G.A., Wanjoya, A.K.: The power of the pruned exact linear time(pelt) test in multiple changepoint detection. Am. J. Theor. Appl. Stat. **4**(6), 581–586 (2015)

Alleviating Congestion in Restricted Urban Areas with Cooperative Intersection Management

Levente Alekszejenkó(✉) and Tadeusz Dobrowiecki

Budapest University of Technology and Economics, Budapest, Hungary
ale.levente@gmail.com

Abstract. Congested urban traffic frequently deviates toward residential areas, where due to local conditions congestion builds up quickly. The increasing traffic flow there causes health, environmental and safety problems. A pragmatic solution is to make these areas infeasible as escape routes. In the following, an area wide cooperative intelligent intersection management system is proposed, aiming at suppressing the number of passing-through vehicles if their density increases too much. The solution is based on communicating traffic light controllers using the analogy of the Explicit Congestion Notification (ECN) protocol and organized as a hierarchical Multiagent System. The system was implemented by extending an open-source traffic simulation tool, so called Eclipse SUMO. Simulations were performed on a simplified model of a residential area. The efficacy of the proposal was evaluated with Macroscopic Fundamental Diagrams of the investigated traffic. The results are promising opening way to further research.

Keywords: Intelligent intersection management · Explicit Congestion Notification (ECN) · Eclipse SUMO

1 Overview of the Problem

Congestion is curse of modern cities, but not all urban areas are evenly affected. Ad hoc escape routes, alternative routes recommended by navigation applications frequently shepherd heavy traffic through less congested streets in residential areas, green recreational areas, areas with educational and health care institution campuses, and the like [1]. If a higher amount of vehicles appears in these areas, which usually have narrower roads with a lower throughput and plenty of pedestrian traffic, congestion builds up quickly.

Congestion in such areas, however, introducing pollution and noise, means a serious degradation of the quality of life conditions, an increased danger to the residential or commuter safety, especially to children, and an increased difficulty for passing of emergency vehicles [1–3]. In addition, extremal weather conditions, due to frequently steep, slippery roads, may negatively influence here already the normal traffic, not to mention it when the congestion sets in [1].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 34–52, 2021.
https://doi.org/10.1007/978-3-030-55180-3_3

Such areas could be protected with traffic code and special technical means (like e.g. limiting, prohibitory, and warning traffic signboards, swinging barriers, congestion charging zones, special bus and HOV lanes, reversible traffic lanes, etc. [4–6].) only in part. First of all the roads here are usually narrow and frequently single-lane or one-way, furthermore in case of mild normal noncongested traffic, traffic regulation should not differ from other parts of town and should permit the righteous drivers to reach their destinations within. Periods of traffic jam mean a real difference, but then the effect of many enforceable traffic rules weakens. Infrastructure development (speed bumps and tables, designed drop off points for delivery, more pedestrian crossings, creation of the residential shared streets, etc.) is of course an option, but also for the budget of the local authority [4].

In the following a review of the literature will be given in Sect. 2. The proposed solution is presented in Sect. 3. The mutual dependence of the intersections is the topic of Sect. 4. The kernel of the solution - the analogy to the ECN protocol, signal optimization, and the agent communication are discussed in Sects. 5, 6, and 7. Simulation platform is introduced in Sect. 8, and simulation measurements are presented in Sect. 9. Finally Sect. 10 concludes the paper.

2 Related Works

Urban congestion is an acute problem perceivable to everybody, no wonder that the literature on this topic is vast. Plenty of ideas have been tried in practice [4, 5], or only as a theoretical consideration, verified in simplified models by simulations [22, 24, 28, 31–34, 45, 48, 53].

One of the most interesting approaches is the so called microscopic simulation where attention is given to the behaviour of individual vehicles and drivers [35, 58–60]. A natural AI (Artificial Intelligence) paradigm, covering it, is a Multiagent System (MAS), with plenty of implemented and published concrete frameworks [17, 20, 25–28, 30, 38, 40, 41, 43, 46, 50, 52, 56, 57, 61, 62] (for a review, see also [19, 23, 44]). Agent-based approach will gain even more importance in the future, because the further development of the traffic technology, the even higher participation of the connected and automated vehicles makes such models indispensable. One must consider in particular that although the vehicle technology is already satisfactorily developed, not so its intelligent software and the automated decision making aspect, which relay heavily on the future acceptance of such every-day automated tools and on the possible emergent new traffic related psychological problems, perhaps even some new forms of antisocial behaviour [49, 51].

Hierarchical MAS systems (where not only individual vehicles are cooperatively organized, but also intersections controlling them) were experimented with already in [17, 20, 21, 25, 26, 38, 40, 41, 52, 62]. In [20, 39, 41, 62] the intersection managers collaborate on the adaptation of signal plans by a consensus, within the reach of the intersection affected by the changes. Sensing congestion is solved by sensing the level of an instant congestion by implemented detectors and averaging over a specified time window. At a level of individual intersections signal protocols can be optimized in a variety of ways [64–72], for review see also [67].

Another issue related to the individual intersections is the organization of the vehicle agents into platoons. Platoons can be formed on various basis: optimized for travelling time and fuel consumption on freeways, based upon the density of connected and non-connected vehicles (with vehicles preferring not to communicate due to privacy or security reasons), organized by gaps in the traffic (with an emphasis on close and steady spacing), forming platoons for cruising and keeping formation by a consensus, multi-modal platooning with an eye on the optimal travel time, etc. [18, 32, 36, 37, 46, 47, 55].

The possibility that the vehicle agents do not behave bona fide was investigated in [22]. What swindling behaviour will be permissible as the interplay between the hard-wired automatic behaviour and the adaptivity to the owner level programming is not yet clear, as the programming issues of the automated vehicles are not yet finalized and resolved [51].

In the analysis of the congestion the congestion alleviation can be paired with computing route deviations [27, 28], or tackling multi-objective problems [57]. There are also attempts to improve the solution by introducing MAS learning, like in [29, 42, 44, 54]. Learning in MAS systems has many forms, depending on the available information and the cooperativeness/competitiveness of agent organization. Urban traffic field is clearly a competitive field where the multiagent learning faces serious theoretical difficulties.

3 Proposed Solution

Our further research into the problem of congestion alleviation in the sheltered areas was based upon the following simplifying (and perhaps realistic) assumptions:

(A1). Congestion comes from the number of vehicles in the traffic, and that cannot be decreased magically on the short notice. Locking out vehicles from an area permanently (pedestrians only zones) does not always work, because the area should be passable at any time by its righteous inhabitants or commuters. Consequently when the danger of congestion becomes perceptible, the number of vehicles permitted to enter and to pass through the area should be adaptively mitigated, on the expense that the surplus of vehicles is pushed out beyond the sheltered area borders. There, on larger roads, a more efficient congestion control may be ventured [4–6].

(A2). The only traffic regulation authority (aside from flesh-and-blood policemen) relatively well observed (even by stressed commuters in congested traffic) are traffic lights at controlled intersections. Here intersection management should be adapted locally to serve the surging traffic, but should also join the common effort (with other intersections) to push out the surplus of the vehicles from the sheltered area. We assume also that the traffic sensing is technically the simplest, based solely on the loop and area detectors, connected directly to the local intersection managers [6]. The proposed algorithmic solution to this problem is the main contribution of this paper.

To solve the area sheltering problem constrained by the above assumptions, we propose thus:

(P1). The organizational paradigm of a hierarchical, 2-layer cooperating multiagent system (MAS), see Fig. 2. This paradigm was used to conceptually clarify and fuse together the natural agent-like characteristics of all the participants in the urban traffic. Basically, we can discern two types of agents.

Smart cars: Smart cars are connected, intelligent, self-driving vehicles (SAE 5th level [7]). They can communicate with each other (V2V) and with the infrastructure as well (V2I). They communicate and cooperate to form platoons near intersections. Please note that in our approach vehicles cooperate if their immediate interests - headings within an intersection - coincide. In consequence the smart car notion may apply - as an abstract model - also to the cooperatively behaving human-driven vehicles as well.

Intersection managers, called for short judges: Judges are intelligent traffic controller agents; every intersection in the network is maintained by one. The main task of the judges is to decide and control which groups (platoons) of vehicles can pass through the intersection at any given time (owning green-light phase). Judges can communicate with each other and with smart cars as well. Please note that the body of the possible cooperating agents and the hierarchy is open (in the future research) also to pedestrians, cyclists, enforcement personnel and the like.

We assume an ideal communication between agents: without packet-loss, and with appropriate transmission times and delays. All agents are trustworthy and bona-fide. They can understand the received messages and can act accordingly.

(P2). The essence of the lower layer in Fig. 2 is the intersection-level cooperation of vehicles for a more efficient passing through. Vehicle agents, intending to pass the intersection in the same heading, self-organize themselves into platoons in the immediate proximity of the intersection and sign up at the intersection manager agent (judge) reporting their travel goals. The batch of platoons drawing near to an intersection from all directions constitutes a set of jobs competing for the common resource, the green light phase. The intersection judge continuously re-designs on this basis an optimized light control program, using an analogy to an operating system job scheduling algorithm (Round Robin and Minimal Destination Distance First). This approach was already reported in [8], illustrated with measurements run on a SUMO-based simulation platform extended with a proprietary overhead to implement agent communication and cooperation [9]. The formation of platoons (proven beneficial in free highway traffic due to a much smoother driving patterns) is applied here to reduce the computational complexity burden of the judges, because this way the communication between judges and smart cars can be reduced.

(P3). The higher layer of the agent system in Fig. 2 embraces intersections defining the sheltered area, by the dependence of their passing through traffic. Agents in this layer may be considered a coalition created with the single goal of keeping the supervised physical environment at a mild level of congestion. The set of so coupled intersections can be selected manually, but can also be determined algorithmically from the road and intersection topology (see Sect. 4).

The congestion alleviation is based on the analogy to the computer network Explicit Congestion Notification (ECN) protocol [10, 11, 16]. Within the designated area congestion is sensed by lane area detectors on approachings to every intersection [6], and if congestion onset is observed, all the neighbour intersection managers being upstream of the incoming traffic are notified (see Fig. 3). The aim is to warn them to make them replan their light control (to shorten the appropriate phases) to abate the culprit flow of the vehicles. The chain of warnings propagates thus upstreams, if necessary, until it reaches the area borders, where the border intersection managers insulate the sheltered

area from the too excessive traffic. Because of the used protocol the managers in this layer are called ECN-judges.

In the following we describe in detail the main components defining the activity of the higher layer of the MAS system.

4 Coupled Intersections

A typical urban area addressed in Sect. 1 can be composed from many intersections, not everyone being equally responsible for building up congestion. As the number of intersections scales up exponentially the state space of the proposed system, it should be beneficial to limit the number of the considered dependent intersection to a functionally necessary minimal count. To this end it is vital to find out which are those intersections, which are "coupled together" by the vehicle flow, i.e. intersections belonging to the main traffic directions. This feature depends greatly on the topology and on the classification of the road network, and should be verified independently for all the directions in the traffic.

Coupled intersections can be picked manually by screening the road topology. However, if all possible routes in an area can be listed (and this is computationally feasible), an algorithm can be designed to find out which intersections should communicate and coordinate. For e.g., if two intersections (say A and B) are always following each other in the same strict order A $<$ B in all routes containing both intersections, without any other intersections between them, then these two intersections are coupled by the traffic flow. By using this approach iteratively (see Appendix A), a set of coupled intersections can be identified. Such set of intersection agents constitutes the higher level of the multiagent system, a coalition of agents united with a common aim to fight away the congestion.

5 Congestion Sensing and Notification - ECN-Judges

To achieve the global aim of area sheltering and to utilize effectively the congestion-related re-calculation of the signal plans intersection managers (judge agents) in the intersection coupling set must solve two additional tasks. One is to identify somehow whether the number of vehicles currently exceeds the predefined limits (at the receiver junction). Then they should spread this information to the responsible intersection managers (at the sender junctions) to adjust their signal plans accordingly. As a result, the number of vehicles between these two junctions will be limited by signal coordination. The idea behind this solution is pretty much the same, as that behind the Explicit Congestion Notification (ECN) protocol of computer networks [10, 16]. Due to this similarity, the proposed new intelligent intersection managers will be referred to as the ECN-Judges. The ECN-based communication between agents is described in detail in Sect. 7.

Sensing the onset of congestion is not easy, as there is no universal and approved definition of what the congestion is [12, 13]. If preliminary traffic measurements for the supervised areas are available a solution may be to measure the occupancy percentage of the roads incoming from the intersections coupled by the one supervised by the actual judge. If this percentage exceeds a certain fraction (e.g. 90% of the occupancy of

the maximal traffic flow) then one can say a congestion is forming at this intersection. If no preliminary measurements are available for comparison, local occupancy can be measured by e.g. lane area sensors [6]. Yet another solution may be to estimate the probability of the congestion [14]. Sensory measurements are the best because the measured lane occupancy (lane area fraction occupied by the vehicles) is a true indication of the problem without resorting to models and to take into account vehicles of various dimensions. In the simulations (Sect. 9) lane area detectors were used.

6 Generating Signal Plans

The state-space of an ECN judge can be enormous since it depends both on the number of the incoming vehicles and on the number of the neighboring intersections. Therefore storing signal plans for all of the states is quite memory consuming. Instead, a dynamical signal plan generation method was designed (for an overview, see Fig. 2). The calculation of a simple signal phase is formally an integer programming problem (IP) (see Appendix B) [15]. Our goal is to maximize the number of directions which receive green light at the same time, subject to the actual state of the roads. This state consists of a dynamic part, like the incoming congestion warnings or the decision of the scheduling algorithm (e.g. Round Robin), as well as static part, which describes which directions cannot go through the intersection simultaneously.

The solution to the IP problem specifies in the intersection which incoming directions are free to pass (green light) and which are stopped (red light). The idea is to momentarily stop directions which possibly contributed to the observed congestion danger downstream.

If the IP problem is solved, we only know the signal phase for a given moment. In order to generate a signal plan (which describes how long a given heading should own a green or a red light), it is necessary to recalculate the solution vector from time to time. In our approach the recalculation time T is a linear function of the number of incoming vehicles, but cannot exceed 40 s:

$$T = \begin{cases} 1.5N_v + 5, & if N_v \leq 23 \\ 40 & else \end{cases} \tag{1}$$

This empirical equation takes into account the number of vehicles currently receiving a green light N_j, and sets an upper bound (40 s) on the signal cycle time. We calculate that in a usual intersection ca. 1.5 s time is needed for a vehicle to pass through, and we add another 5 s to account for the starting of the first vehicle and the clearing of the last vehicle of the passing vehicle queue.

7 Agent Communication

ECN-judge agents are able to estimate the occupancy of the supervise lanes. Every 15 s they check the congestion state of the incoming lanes from the intersections belonging to the coupled intersection set. If the possibility of congestion is indicated, judge agents broadcast warning (ECN-message), otherwise they broadcast that everything is

in order. That way we can ensure that the knowledge-bases of the coupled ECM-judges remain consistent. If there are two neighbouring intersections in the coupling set having a congested road section in-between, the judge agent responsible for injecting too many vehicles into the section will - after at most a 40 s delay - limit the flow of vehicles towards the warning agent.

Congestion warning (ECN-message) should truly be an event-based message. Periodic broadcast is a simplifying solution in our case. However, in the case of real communication channels, periodic broadcast provides better fault-tolerance. Periodic broadcast automatically ensures the consistency of the local knowledge-bases of the judge agents, while in the case of an event-based broadcast specialized algorithms would be required. Periodic broadcast can serve also as a kind of "life sign", indicating that an agent remains active, even if it momentarily has nothing to do. Moreover the communication channel overload is not high, considering the low frequency and short information content of the messages. Depending on the infrastructure the channel can be even realized as a point-to-point connection, not involving the wireless communication.

An ECM-message contains the ID of the sending ECN-judge agent, the ID of the road segment leading to the sender from the coupled intersection area, and the required status of the road segment (i.e. congested or not). Upon receiving such message the addressed ECN-judges store its status content and use it later when the re-calculation of the signal plans is scheduled.

Agent-agent communication is also the basis of the cooperation realized at the lower level of the agent system (see Fig. 2) [9]. Here vehicle-to-vehicle (V2V) communication is used to organize platoons, and vehicle-to-infrastructure (V2I) communication (the communication between platoon leader agents and intersection judges) is used to negotiate the most effective passage through the intersection.

8 Simulation Platform

The proposed system was validated by using a simulation platform based on Simulation of Urban MObility (SUMO) [63]. This platform was already extended in [9] with a proprietary code realizing tools to implement perception-decision making-action loop of the vehicle and intersection agents, together with the provision for the inter-agent communication. The choice of the SUMO platform with agent-like overhead instead of a truly agent-based design was dictated by better quality indicators when the number of concurrently simulated agents was at stake [26].

9 Measurements and Results

To verify the merits of the proposed approach, a simplified abstract residential areas were created, see Fig. 4. Thick lines designate arteries suitable for a high throughput traffic flow (3 lanes in each direction). These roads surround an area where only narrow streets exist: (a) a single thin diagonal, single lane road, with a web of possibly dead-end side roads (not included), or (b) a network of similarly narrow roads enmeshing the area. With a single diagonal residential road we attempt to model an attractive escape road to the commuters, and with a more complicated road system, the residential web of smaller

streets and alleys. Traffic demand was injected at the lower-left intersection toward the upper-right intersection and vice versa. As a basic load 360 vehicles were injected in every direction, within 1800 s time, in a uniformly random manner. This traffic demand was upscaled (Scale N = load × N), with the maximum vehicle flow being ten times of the original value.

The travel goal of every vehicle was set to lower-left, or to upper-right corner accordingly (depending on the travelling direction). In the simulation in the test environment (a) vehicles were pre-programmed to turn randomly into the internal area (1/3) or travel around it (2/3). We attempted to model this way a number of commuters returning home. In the second simulations in the test environment (b) the traveling trajectory was to be chosen freely. Upon injecting vehicles into the simulator, SUMO directed every vehicle along the trajectory considered the fastest in a given moment (Dijkstra algorithm). We assume that such mechanism imitate well the advices provided by the navigation applications used by the drivers.

The inner intersections (at both ends of the diagonal (a), or at all intersections within (b)) were equipped with ECN-judges. Simulations compared the traditional signaling system (built-in into the SUMO) with the effectiveness of the signal planning based on the ECN, proposed in the paper. Simulations were run until all injected vehicles left the network (at the other end) and various statistics of the traffic within and without were measured under variable load conditions.

The coupling set calculation was trivial in test (a). In test (b) the coupling set was established manually. The proposed algorithm was used in modeling of a highly irregular Budapest traffic node [73]. The algorithm was included here, however, because it constitutes an important part of the approach, for more involved situations.

The evolution of the traffic was measured and compared based on so called Macroscopic Fundamental Diagram (MFD), see Fig. 5 [14]. An MFD presents the evolution of the flow of the vehicles (usually in vehicle/h) as a function of the density of the vehicles (usually in vehicle/km). Low density (left) part of an MFD reflects a free (non-congested) flow of vehicles, with the maximum permissible speed, and shows usually little variation. It reaches the top flow value at a critical density where the style of driving changes due to closer distances between the vehicles. From here the right part of the diagram reflects an even more congested behaviour, with growing density and decaying flow, until the full standstill of a jam. Congested behaviour is less stable and shows a much more variability of the flow and density values.

Measurement results measured with the traditional signaling system (and non-cooperative intersection managers) and the ECN-judge controlled solution are shown in Fig. 6 and 7. As it can be seen in the MFD in Fig. 6, in test (b) there is no much difference between the two approaches, and not much sensitivity to the measure of the traffic overload. The culprit here may be the freedom of the navigation applications to re-direct vehicles from without to within the area, once the inner area becomes temporarily easier to the traffic, with plenty of alternative, almost as fast routes.

In contrary, there is a pronounce difference in test (a), where due to the assumed blind streets, navigation may only push the traffic through the diagonal. Here, as we interpret it, owning to the work of the ECN-judges, free flow can be maintained longer (as the function of the traffic overload), and even under heavy overload the proposed

solution provides for faster flows and/or higher density of the still not fully congested traffic.

As the action of the ECN-judges forces down the number of vehicles passing the intersections, injected vehicles leave the simulated area later, than in the traditional case. Comparing, however, to a constant level of congestion visible in the overloaded traditional case (see Fig. 7, density measured as the occupancy over a lane area detectors), their density fluctuates. If this density jumps higher, the bursts of the ECN-messages introduce a feed-back control, reducing it considerably. The observed surges in density are probably caused by the temporarily full availability of the lanes and its sucking effect on the vehicles and a possibly later pilling up at the red lights.

10 Conclusions

Urban vehicle traffic is a physical system of enormous complexity. At the dawn of mass usage of the connected automated vehicles and related subjective (and irrational?) human decision making we may expect that this complexity will intensify even further, bringing issues difficult to understand and to solve. Clearly no single technical solution can address these problems on the whole. More and more ideas must be deployed, tried, integrated, verified under rich variety of practical conditions. We see our approach as an element of this trend.

Limiting the amount of vehicles in certain regions of the road network can be a desirable, yet a challenging task. The traffic signal coordination system, proposed in this paper, shows promising results and might add to solve this problem. The proposed system was designed and verified at a relatively high abstract level. Evaluating its further merits may depend on the details of the physical implementation, on the particulars of the infrastructure and the environment where such system may be tried, and also on what additional elements of intelligence and optimal decision making may be feasible to add to.

Works the closest to our approach are perhaps [20, 39, 41, 62]. There intersection managers collaborate on the adaptation of signal plans by a consensus, within the reach of the intersection affected by the changes. In our approach the affected intersections depend on the borders of the sheltered area (which is defined by its particular character) and the direction of the interactions is specifically based on the flow dependency from downstream to upstream, making the communication a less burden to the agents. Sensing congestion is solved similarly by sensing the level of an instant congestion by implemented detectors and averaging over a specified time window (using a moving average in our case). The principal novelty is founding the cooperation of the intersection managers layer on the analogy to the effective but simple ECN congestion eliminating protocols from the computer networks.

Another issue related to the individual intersections are vehicle platoons. Our solution is closely related to [18, 46], where existing queues and significant platoons approaching each intersection are identified and their parameters estimated from the measurements. In our approach, however, platoons are not identified by their density, but consciously negotiated close to intersections, intensively using V2V and V2I communication options. Platoons are formed by a short term joined interest of heading toward the same exit in

the intersection and this is reported to the intersection managers for decision making. With this model we can cover automated and human-driven vehicles also, as close to an intersection even an independent human driver will file into a lane consistent with her/his heading, and a covert dialogue of her/his intelligent vehicle with the intersection will serve her/his traveling aims as well.

In our work we tacitly assumed that the congestion is the primary problem, every other objective (like air pollution, convenience, pedestrian road safety, bad weather safety, serviceability, fuel consumption, enjoyment value, etc.) is derivative of the congestion. Once congestion is alleviated, all other problems will also be reduced. A singular objective - alleviating congestion - simplifies control and computation.

In the future we intend to investigate the interaction between the congestion formation and alleviation and the problem of an effective parking. Parking in residential areas is an interesting problem, because such areas act (especially in time of morning or evening congestions) as a sink or a source to a part of the vehicle flow. Parking is far from being a simple and homogenous problem. It depends upon the intended place (P + R, gated community, shopping mall, office deep garage, highway, drop-off points, etc.), the aim (short term, for shopping and leisure, for working hours, for the night), the identity of the parking decision making agents, and on the demands posed by other agents present in the parking area. We also expect that the setting of the problem and the possible solutions will be seriously influenced by the composition of the vehicle fleet form various SAE vehicle classes [7].

Acknowledgments. The research has been supported in part by the European Union, cofinanced by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications), and in part by the BME- Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC).

Appendix

A. Searching for a Set of Strongly Coupled Intersections

In road networks, defining intersection dependence is not trivial. In e.g. American-like, rectangular road networks at least two independent routes between any points of the network may exist, while European-like, irregular networks may contain bottlenecks, i.e. sections which are used by the whole traffic flow between two points of the network. These points or intersections are coupled somehow together, and the whole road between such coupled intersections behaves similarly to the conflict zone of a single intersection. Vehicles are competing for the usage of this road surface. For this reason, it might be beneficial to avoid congestion and therefore to limit the amount of conflicts along these road segments.

The goal is thus to identify the maximal group of coupled intersections. The algorithm accepts as input road network (topology) (N), every possible route through this network, given by an ordered list of the visited junctions (R), the highest distance bound between the coupled intersections (Lmax) and a given intersection (a), which coupled intersections are to be found. The algorithm runs like follows:

1st step: X := {a}
2nd step: x : = an intersection in N, which has an out going route to an intersection y, already in X and the distance d(x,y) < Lmax.
3rd step: K : = {every route from R, containing road segment x → y}
4th step: **If** there is no element in K, containing both x and y, and any other intersection from N in-between, **then** X : = X ∪ {x}
5th step: **If** there is another applicable intersection in N, **then go to** 2nd step **Otherwise:** Return X

The returned X set contains intersections coupled with the originally selected (a) intersection. Whole sections of the road network between such intersections behave similarly to the conflict zones of the traditional intersections. In order to reduce the number of possible conflicts, coupled intersections should share the congestion notification between each other.

As an example consider Fig. 1. Left turns are prohibited at two junctions. At B, moving from A C is prohibited. Similarly, at G, moving from F towards E is also prohibited. The question is which junctions are coupled to junction H?

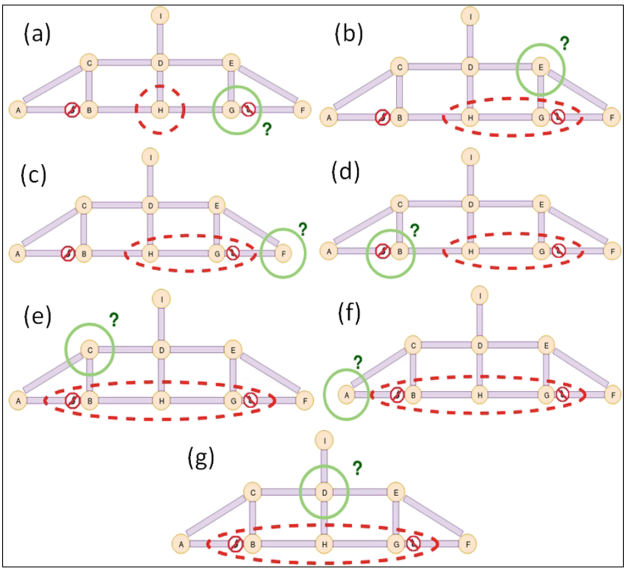


Fig. 1. Collecting coupled intersections.

With the proposed algorithm we found that G is a part of the coupling set of H, since due to the prohibition of the left turn, every route, containing G → H, will contain H directly after G. Let us see now the case of E, a neighbor of G. Because there are two possible routes from E to G (one direct route, and one via intersection F), E is not a part of the coupling set of H. The same also holds for F, therefore it does not belong to the set either. As the example network is symmetric, we can accept also B as a part of the coupling set of H, but not C and A.

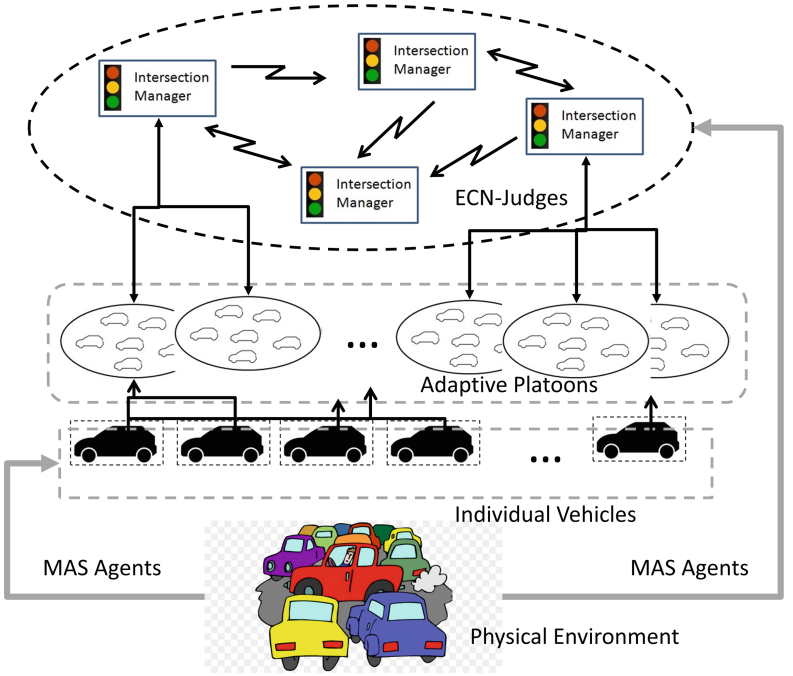


Fig. 2. General setup of the congestion alleviating MAS system (lower cartoon by www.freepik.com, last accessed 2020/02/27.)

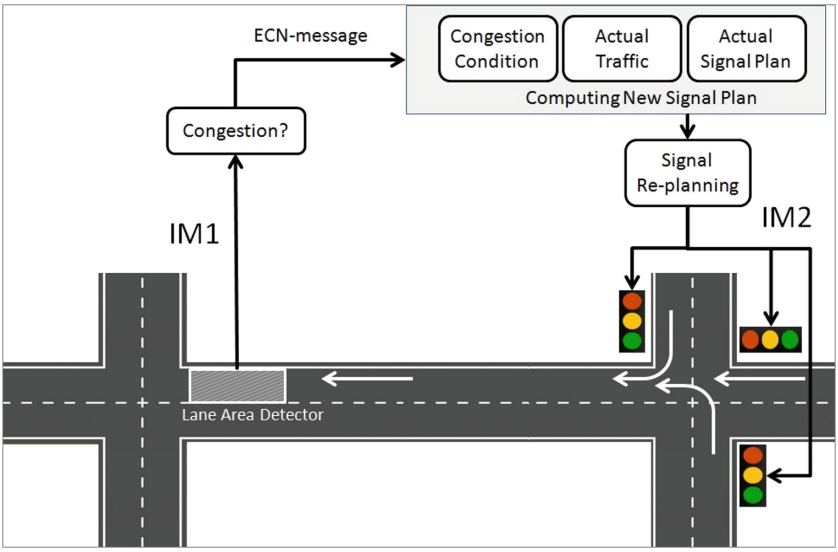


Fig. 3. Schematic view of the intersection cooperation based on the ECN warning.

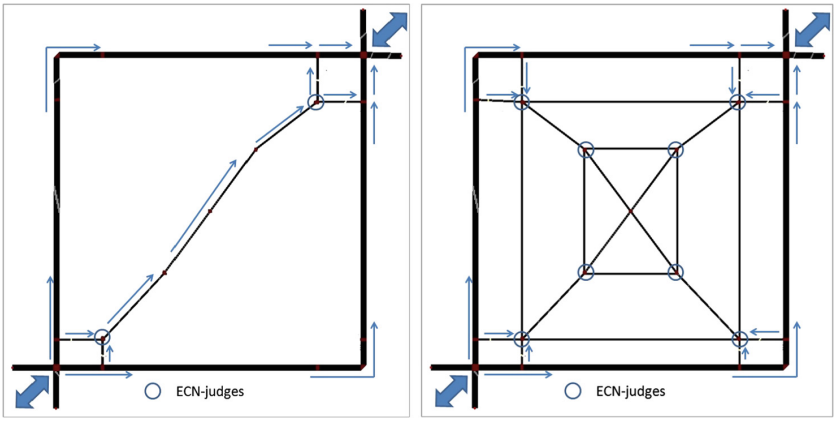


Fig. 4. Test environments representing residential areas: (a) a single attractive escape route with a web of blind alleys, (b) a web of passable residential streets. The traffic flow is bidirectional, but only the single direction flow is shown for simplicity. ECN-judges are shown with circles.

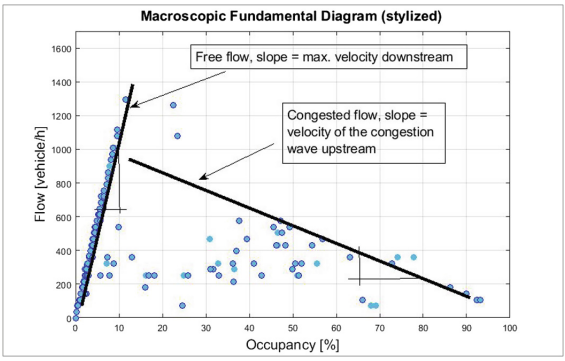


Fig. 5. Flow vs. Density Macroscopic Fundamental Diagram (MDF). Left side belongs to the (stable) non-congested free flow, with the maximal (permissible) speed of driving, until the critical density is reached (the top of the diagram). Right side shows the formation of an unstable congested flow, with an ever increasing density and drop in velocity, until the traffic stops entirely at the jam density. In our measurements the % occupancy of the area loop detectors was taken as the (equivalent) measure of the traffic density.

The last intersection to be investigated is D. Due to the many possible routes between them (e.g., DH, DC(A)B, DE(F)G), D have to be left out from the set.

B. Signal Phase Generation as an Integer Programming

Our simulation platform receives a two-dimensional matrix, the so-called conflict-matrix **C** as a configuration input. This matrix describes which directions may not pass through the intersections simultaneously. (If allowed, it would cause a risk of a collision.) $[\mathbf{C}]_{i,j}$

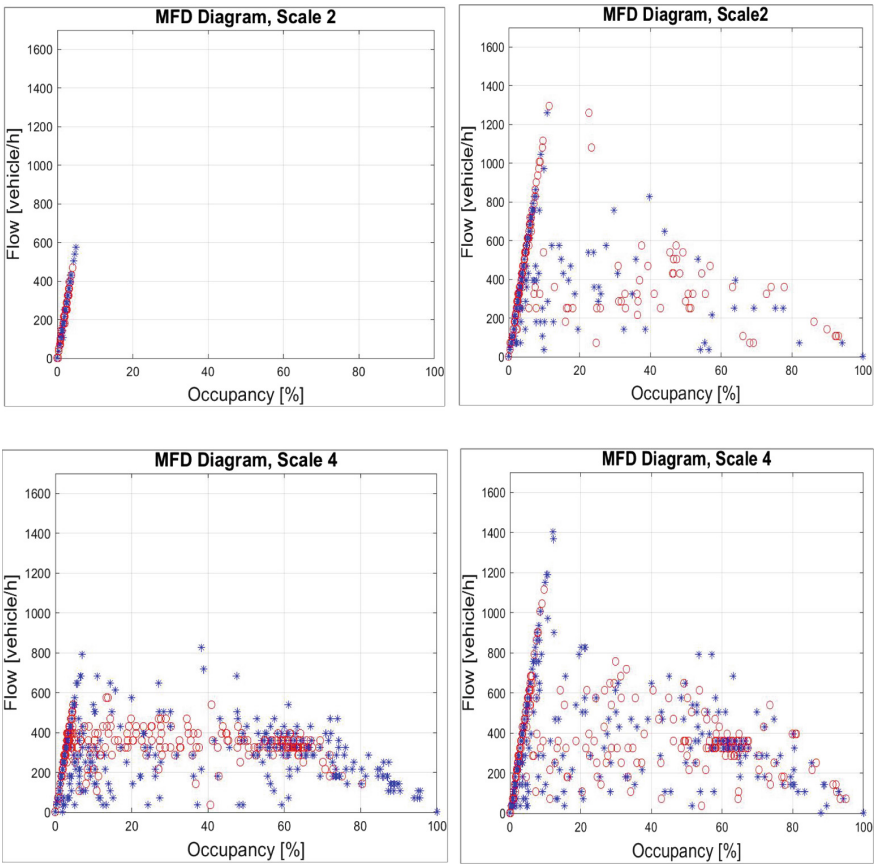


Fig. 6. MFD diagram showing the effect of ECN-based traffic limitations in the test environments. Test (a) left column, test (b) right column (* - ECN, o - traditional solution).

element is 1 if the *i*th and *j*th directions are prohibited to move simultaneously, and is 0 otherwise.

The principal goal of signal phase generation is to provide green lights for as many vehicles as safely possible. Some other constraints also have to be taken into consideration, like the scheduling decision, or the traffic reduction indicating by the incoming ECN-message, proposed in this paper.

This problem can be formalized as an integer programming problem (IP). Let us define vector **x** as the vector of directions. The x_i coordinates of **x** will be numerically constraint to be 0 or 1, indicating that the *i*th direction can or may not receive a green light in the current phase. Naturally our goal function is to maximize the L_1-norm of **x**, i.e. max $\Sigma\ |x_i|$.

The constraint matrix of the IP problem can be formed as follows. Let us iterate through the conflict matrix **C**. If a specific $[\mathbf{C}]_{i,j}$ element equals 1, we add a new constraint: $x_i + x_j \leq 1$. These constraints (and their integer solution) ensure that at most one

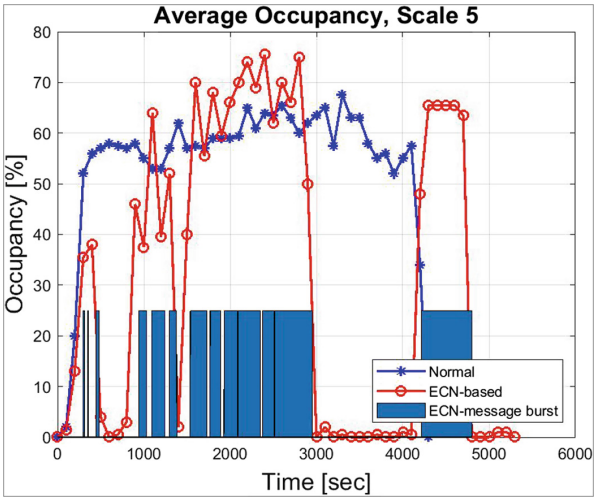


Fig. 7. Congestion alleviating effect of the ECN-messages in the test environment (a). Traditional signal control (blue), ECN-based control (red).

of the conflicting directions will receive a green light. Moreover, we shall add an $x_\mathrm{i} \leq 1$ for every ith coordinate. This two types of constraints warrant that every x_i coordinate will fall in the range [0,1], but the integer solution will warrant that every x_i will equal either 0 or 1.

For the kth direction which must receive a green light, due to the scheduling decision, an $x_\mathrm{k} = 1$ constraint will be added. Similarly, when an l direction must not receive green light, in effect of an ECN-signal, $x_\mathrm{l} = 0$ constraint will also be added.

The solution of this problem (2) will provide an optimal configuration of the traffic lights. The constraints also prevent conflicting directions to receive green lights simultaneously.

$$max \sum_i |x_i| \tag{2}$$

subject to:

$$x_{i_1} + x_{j_1} \leq 1, x_{i_2} + x_{j_2} \leq 1, \ldots, x_{i_m} + x_{j_m} \leq 1 \qquad \text{(constraints)}$$

$$x_k = 1 \qquad \text{(scheduling)}$$

$$x_{l_1} = 0, x_{l_2} = 0, \ldots, x_{l_n} = 0 \qquad \text{(ECN-notification)}$$

References

1. Macfarlane, J.: Your Navigation app is making traffic unmanageable. IEEE Spectr. 22–27, 19 Sep 2019
2. Swift, P., Painter, D., Goldstein, M.: Residential Street Typology and Injury Accident Frequency, Congress for the New Urbanism, Denver, Co., June 1997
3. Elevated Traffic Incidents in Long Beach School Zones, Long Beach Area News 3 April 2014. https://www.lbrag.com/2014/04/elevated-traffic-incidents-in-long-beach-school-zones/1132/. Accessed 27 Feb 2020
4. How to Fix Congestion, Texas A&M Transp. Inst. https://policy.tti.tamu.edu/congestion/how-to-fix-congestion/. Accessed 27 Feb 2020
5. Meyer, M.D.: A Toolbox for Alleviating Traffic Congestion and Enhancing Mobility. Inst. of Transp. Eng., Washington, DC (1997). https://rosap.ntl.bts.gov/view/dot/2145. Accessed 27 Feb 2020
6. Traffic Control Systems Handbook: Ch 6. Detectors, U.S. Dept of Transp., Federal Highway Adm., Pub. No. FHWA-HOP-06-006, Oct 2005. https://ops.fhwa.dot.gov/publications/fhwahop06006/chapter_6.htm. Accessed 27 Feb 2020
7. SAE Standards News: J3016 automated-driving graphic update. https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic. Accessed 27 Feb 2020
8. Alekszejenkó, L., Dobrowiecki, T.: SUMO based platform for cooperative intelligent automotive agents. In: EPiC Series in Computing, vol. 62, pp. 107–123 (2019)
9. Alekszejenkó, L., Dobrowiecki, T.: Intelligent vehicles in urban traffic – communication based cooperation. In: IEEE 17th World Symposium on Applied Machine Intelligence (SAMI), pp. 299–304 (2019)
10. Floyd, S., Ramakrishnan, K., Black, D.L.: The Addition of Explicit Congestion Notification (ECN) to IP, Sept 2001. https://tools.ietf.org/html/rfc3168#section-1. Accessed 27 Feb 2020
11. Gomez, C.A., Wang, X., Shami, A.: Intelligent active queue management using explicit congestion notification. In: IEEE Global Communication Conference – GLOBECOM, Waikoloa, HI, USA (2019)
12. Aftabuzzaman, M.: Measuring traffic congestion - a critical review. In: Proceedings of the 30th Australasian Transport Research Forum (2007)
13. Rao, M., Rao, K.R.: Measuring urban traffic congestion – a review. Int. J. Traffic Trans. Eng. (IJTTE) **2**(4), 286–305 (2012)
14. Estel, A.: Quality of Service Beyond the Traditional Fundamental Diagram. In: 75 Years of the Fundamental Diagram for Traffic Flow Theory Greenshields Symposium, Woods Hole, Massachusetts, 8–10 July 2008, pp. 86–106 (2008)
15. Bradley, S., Hax, A., Magnanti, T., Applied Mathematical Programming, Ch 9. Integer Programming. Addison-Wesley (1977)
16. Lu, Y., Fana, X., Qian, L.: Dynamic ECN marking threshold algorithm for TCP congestion control in data center networks. Comput. Commun. **129**, 197–208 (2018)
17. Wenkstern, R.Z., Steel, T.L., Daescu, O., Hansen, J., Boyraz, P.: Matisse: a large-scale multi-agent system for simulating traffic safety scenarios. Ch 22. In: Hansen, J., et al. (eds.) Digital Signal Processing for In-Vehicle Systems and Safety. Springer, New York (2012)
18. Bashiri, M., Fleming, C.H.: A platoon-based intersection management system for autonomous vehicles. In: IEEE Intelligent Vehicles Symposium(IV), Redondo Beach, CA, USA, 11–14 June 2017, pp. 667–672 (2017)
19. Chen, B., Cheng, H.H.: A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transp. Syst. **11**(2), 485–497 (2010)
20. Torabi, B., Wenkstern, R.Z., Saylor, R.: A self-adaptive collaborative multi-agent based traffic signal timing system. In: IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 16–19 September 2018

21. Beak, B., Larry, H.K., Feng, Y.: Adaptive coordination based on connected vehicle technology. Transp. Res. Rec. J. Transp. Res. Board **2619**(1), 1–12 (2017)
22. Zhang, X., Wang, D.: Adaptive traffic signal control mechanism for intelligent transportation based on a consortium blockchain. IEEE Access **7**, 97281–97295 (2019)
23. Kesting, A., Treiber, M., Helbing, D.: Agents for traffic simulation, Ch. 11 In: Multi-Agent Systems: Simulation and Applications. Uhrmacher, A., Weyns, D. (eds.) (2009). arXiv:0805.0300v1 [physics.soc-ph]. 2 May 2008. https://arxiv.org/abs/0805.0300. Accessed 27 Feb 2020
24. Ren, Y., Wang, Y., Yu, G., Liu, H., Xiao, L.: An adaptive signal control scheme to prevent intersection traffic blockage. IEEE Trans. Intelligent Transp. Syst. **18**(6), 1519–1528 (2017)
25. Namoun, A., Marín, C.A., Saint Germain, B., Mehandjiev, N., Philips, J.: A multi-agent system for modelling urban transport infrastructure using intelligent traffic forecasts. In: Mařík, V., Martinez Lastra, J.L., Skobelev, P. (eds.): HoloMAS 2013. LNAI, vol. 8062, pp. 175–186. Springer (2013)
26. Torabi, B., Wenkstern, R.Z., Al-Zinati, M.: An agent-based micro-simulator for ITS. In: 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii, USA, 4–7 November 2018, pp. 2556–2561 (2018)
27. Alqurashi, R., Altman, T.: Hierarchical agent-based modeling for improved traffic routing. Appl. Sci. **9**, 43–76 (2019)
28. McBreen, J., Jensen, P., Marchal, F.: An agent based simulation model of traffic congestion. In: Proceedings of the 4th Workshop on Agents in Traffic and Transportation, Hakodate, May 2006, pp. 43–49 (2006)
29. Mannion, P., Duggan, J., Howley, E.: An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In: McCluskey, T., Kotsialos, A., Müller, J., Klügl, F., Rana, O., Schumann, R. (eds.) Autonomic Road Transport Support Systems. Autonomic Systems. Birkhäuser, Cham (2016)
30. Han, Z., Zhang, K., Yin, H., Zhu, Y.: An urban traffic simulation system based on multi-agent modeling. In: 27th Chinese Control and Decision Conference (CCDC), pp. 6378–6383 (2015)
31. Olia, A., Abdelgawad, H., Abdulhai, B., Razavi, S.N.: Assessing the potential impacts of connected vehicles mobility, environmental, and safety perspectives. J. Intell. Transp. Syst. Tech. Plann. Oper. **20**(3), 229–243 (2016)
32. Khan, S. M.: Connected and automated vehicles in urban transportation cyber-physical systems, Ph.D. Diss., Clemson University (2019). https://tigerprints.clemson.edu/all_dissertations/2475. Accessed 27 Feb 2020
33. Ban, X., Li, W.: Connected vehicle based traffic signal optimization. Report C2SMART Center, USDOT Tier 1 Univ. Transp. Center, April 2018
34. Khan, S.M., Chowdhury, M.: Connected Vehicle Supported Adaptive Traffic Control for Near-congested Condition in a Mixed Traffic Stream, 14 Jul 2019. arXiv:1907.07243 [eess.SP]
35. Fiosins, M., Friedrich, B., Görmer, J., Mattfeld, D., Müller, J.P., Tchouankem, H.: A multiagent approach to modeling autonomic road transport support systems. In: McCluskey, T.L., et al. (eds.) Autonomic Road Transport Support Systems. Autonomic Systems, pp. 67–85. Springer, Cham (2016)
36. Zhou, S., Seferoglu, H.: Connectivity-aware traffic phase scheduling for heterogeneously connected vehicles. In: CarSys 2016: Proceedings of the First ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services, pp. 44–51, October 2016
37. Wu, J., Wang, Y., Wang, L., Shen, Z., Yin, C.: Consensus-based platoon forming for connected autonomous vehicles. IFAC PapersOnLine **51–31**, 801–806 (2018)
38. Srinivasan, D., Choy, M.C.: Cooperative multi-agent system for coordinated traffic signal control. IEE Proc. Intell. Transp. Syst. **153**(1), 41–50 (2006)

39. Cano, M.-D., Sanchez-Iborra, R., Garcia-Sanchez, F., Garcia-Sanchez, A.-J., Garcia-Haro, J.: Coordination and agreement among traffic signal controllers in urban areas, ICTON 2016, Tu.A6.3
40. Abdoos, M., Mozayani, N., Bazzan, A.L.C.: Holonic multi-agent system for traffic signals control. Eng. Appl. Artif. Intell. **26**(5–6), 1575–1587 (2013)
41. Torabi, B., Wenkstern, R.Z., Saylor, R.: A collaborative agent-based traffic signal system for highly dynamic traffic conditions. In: 21st International Conference on Intelligence Transportation Systems (ITSC), Maui, Hawaii, USA, 4–7 November 2018
42. Kuyer, L., Whiteson, S., Bakker, B., Vlassis, N.: Multiagent reinforcement learning for urban traffic control using coordination graphs. In: Daelemans, W., et al. (eds.): ECML PKDD 2008, Part I, LNAI, vol. 5211, pp. 656–671. Springer (2008)
43. Valente, J., Araujo, F., Wenkstern, R.Z.: On modeling and verification of agent-based traffic simulation properties in alloy. Int. J. Agent Technol. Syst. **4**(4), 38–60 (2012)
44. Bazzan, A.L.C.: Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Auton. Agent. Multi-Agent Syst. **18**, 342 (2009)
45. Nakanishi, H., Namerikawa, T.: Optimal traffic signal control for alleviation of congestion based on traffic density prediction by model predictive control. In: Proceedings of the SICE Annual Conference, Tsukuba, Japan, 20–23 September 2016, pp. 1273–1278 (2016)
46. He, Q., Larry Head, K., Ding, J.: PAMSCOD platoon-based arterial multi-modal signal control with online data. Transp. Res. Part C **20**, 164–184 (2012)
47. Heinovski, J., Dressler, F.: Platoon formation optimized car to platoon assignment strategies and protocols. In: IEEE Vehicular Networking Conference (VNC) (2018)
48. Steingraover, M., Schouten, R., Peelen, S., Nijhuis, E., Bakker, B.: Reinforcement learning of traffic light controllers adapting to traffic congestion. In: BNAIC 2005, Proceedings of the 7th Belgium-Netherlands Conference on Artificial Intelligence, Brussels, 17–18 October 2005
49. Brooks, R.: The Big Problem with Self-Driving Cars is People. IEEE Spectrum 27 July 2017. https://spectrum.ieee.org/transportation/self-driving/. Accessed 27 Feb 2020
50. Bazzan, A.: A distributed approach for coordination of traffic signal agents. Auton. Agents Multi-Agent Syst. **10**, 131–164 (2005)
51. Bazzan, A., de Oliveira Boschetti, D., Klügl, F., Nagel, K.: To adapt or not to adapt – consequences of adapting driver and traffic light agents. In: Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning, 5th, 6th, and 7th European Symposium, ALAMAS 2005–2007 on Adaptive and Learning Agents and Multi-Agent Systems, Revised Selected Papers, January 2007
52. Tchappi Haman, I., Kamla, V.C., Galland, S., Kamang, J.C.: Towards an multilevel agent-based model for traffic simulation. In: The 6th International Workshop on Agent-based Mobility, Traffic and Transp. Models, Methodologies and Applications (ABMTRANS), Procedia Computer Science 109C, pp. 887–892 (2017)
53. Orosz, G., Wilson, R.E., Stépán, G.: Traffic jams: dynamics and control. Phil. Trans. R. Soc. A **368**, 4455–4479 (2010)
54. Abdoos, M., Mozayani, N., Bazzan, A.: Traffic light control in non-stationary environments based on multi agent Q-learning. In: 14th International IEEE Conference *On* Intelligent Transportation System (ITSC), Washington, DC, USA, 5–7 October 2011
55. Guoa, Q., Lib, L., Bana, X.: Urban traffic signal control with connected and automated vehicles. A survey. Transp. Res. Part C **101**, 313–334 (2019)
56. Timóteo, I.J.P.M., Araújo, M.R., Rossetti, R.J.F., Oliveira, E.C.: Using TraSMAPI for the assessment of multi-agent traffic. Prog. Artif. Intell. **1**, 157–164 (2012)

57. Jin, J., Ma, X.: A multi-objective agent-based approach for road traffic controls: application for adaptive traffic signal systems. Postprint of Paper VI: Jin, J. and Ma, X. (2018). A multi-objective agent-based approach for road traffic controls: application for adaptive traffic signal systems, under review. https://www.diva-portal.org/smash/get/diva2:1205233/FULLTEXT01.pdf. Accessed 27 Feb 2 020
58. Nagel, K., Wagner, P., Woesler, R.: Still flowing: approaches to traffic flow and traffic jam modeling. Oper. Res. **51**(5), 681–710 (2003)
59. Treiber, M., Kesting, A.: Traffic Flow Dynamics. Springer, Heidelberg (2013)
60. Álvarez López, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.:: Microscopic Traffic Simulation using SUMO. In: 21st International Conference on Intelligent Transportation System (ITSC), pp. 2575–2582 (2018)
61. Dresner, K., Stone, P.: A multiagent approach to autonomous intersection management. J. Artif. Intell. Res. **31**, 591–656 (2008)
62. France, J., Ghorbani, A.A:: A multiagent system for optimizing urban traffic. IEEE/WIC International Conference on Intelligent Agent Technology, IAT 2003 (2003)
63. SUMO Tutorials, https://sumo.dlr.de/docs/Tutorials.html. Accessed 27 Feb 2020
64. Cesme, B.: Self-organizing traffic signals for arterial control. Ph.D. Diss., Northeastern Univ., Boston (2013)
65. Gershenson, C., Rosenblueth, D.A.: Self-organizing traffic lights at multiple-street intersections. Complexity **17**(4), 23–39 (2012)
66. de Oliveira Boschetti, D., Bazzan, A., Lesser, V.: Using cooperative mediation to coordinate traffic lights: a case study. In: 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, The Netherlands, 25–29 July 2005
67. Chen, L., Englund, C.: Cooperative intersection management: a survey. IEEE Trans. Intell. Transp. Syst. **17**(2), 570–586 (2016)
68. Zhihui, L., Qian, C., Yonghua, Z., Rui, Z.: Signal cooperative control with traffic supply and demand on a single intersection. IEEE Access **6**, 54407–54416 (2018)
69. Girianna, M., Benekohal, R.F.: Dynamic signal coordination for networks with oversaturated intersections. Transp. Res. Record **1811**(1), 122–130 (2002)
70. Wagner, P., Alms, R., Erdmann, J., Flötteröd, Y.-P.: Remarks on traffic signal coordination. In: EPiC Series in Computing, vol. 62, pp. 244–255 (2019)
71. Goldstein, R., Smith, S.F.: Expressive real-time intersection scheduling. In: AAAI, vol. 33, pp. 9882–9883, July 2019
72. Feng, Y., Zamanipour, M., Head, K.L.: Connected vehicle-based adaptive signal control and applications. Transp. Res. Rec. J. Transp. Res. Board **2558**(1), 11–19 (2016)
73. Alekszejenkó, L.: Intelligent urban traffic systems. Explicit congestion notification based signal coordination, student competition report. Budapest University of Technology and Economics, Faculty of Electronic and Informatics (2019). http://tdk.bme.hu/VIK/intel/Intelligens-varosi-kozlekedesi-rendszerek. Accessed 27 Feb 2020

Semantic Segmentation of Shield Tunnel Leakage with Combining SSD and FCN

Yadong Xue(✉), Fei Jia, Xinyuan Cai, and Mahdi Shadabfare

Key Laboratory of Geotechnical and Underground Engineering of Education Ministry and Department of Geotechnical Engineering, Tongji University, Shanghai, China
yadongxue@126.com

Abstract. With the rapid development of maintenance of the urban metro tunnel, the structural defects of the metro shield tunnel, especially the water leakage, need to be recognized quickly and accurately. Mask R-CNN is one of the state-of-the-art instance segmentation methods, which has achieved the automatic segmentation of shield tunnel leakage. Although the error rate of the Mask R-CNN algorithm is very low due to a series of complex network structures such as feature pyramid network (FPN) and region proposal network (RPN), the inference cost is 3.24 s per image. Because the structural inspection usually takes only 2–3 h, quick processing of defect images seems necessary. Inspired by a real-time detection method called Single Shot MultiBox Detector (SSD) and the generation of Mask R-CNN, this study constructed a novel convolutional network for fast detection and segmentation of the water leakage. Taking into account the unique appearance and features of water leakage area, it was divided into five groups of different backgrounds to evaluate the interference caused by the complex background and its various shapes. Finally, 278 images were used to test the network, and the average IOU was found as 77.25%, which was close to that of Mask R-CNN. Additionally, the average segmentation time was calculated as 0.09 s per image, far less than Mask R-CNN, which meets the actual requirement of engineering.

Keywords: Shield tunnel · Defect segmentation · Single Shot Multibox Detector · Fully convolutional network · Semantic segmentation

1 Introduction

Along with the development of urban rail transit, the focus of urban rail engineering is gradually changing from “construction-oriented” to “construction and maintenance-oriented”. Due to the complexity of geological conditions and urban underground environment, the shield tunnel has been exposed to many defects, among which the water leakage is one of the most important factors affecting its structural safety. Therefore, it is necessary to check the tunnel regularly. At present, the commonly used methods are based on the nondestructive evaluation, especially on computer vision approaches.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 53–61, 2021.
https://doi.org/10.1007/978-3-030-55180-3_4

Convolutional Neural Network (CNN), which can complete feature extraction automatically in the training stage and perform object classification in the testing stage, has obtained great achievements in the field of object classification. Xue and Li [1] utilized Faster R-CNN (5FPS) [2] for defections of shield tunnels. Their results showed that CNN could extract features automatically and providing outstanding performance for water leakage detection. Inspired by the powerful feature extraction ability of CNN, Long et al. [3] proposed a fully convolutional neural network (FCN) to achieve high-precision semantic segmentation, which is trained pixel to pixel and exceeds the second place on PASCAL VOC dataset by about 20% points. Huang and Li [4] accurately obtained the leakage pixels in the simple background by means of FCN and reported the high performance of FCN in achieving the semantic segmentation of the water leakage in a simple condition.

In spite of the flexibility and capability of FCN, its inherent spatial invariance fails to take account of the useful global context information [5]. This issue may cause the algorithm to ignore the small object instance, which can hinder its application in some problems. Therefore, the current state-of-the-art object segmentation model is based on a two-stage proposal-driven mechanism [6] with the first stage (the backbone) to generate a large number of coarse candidate anchors and the second stage (the CNN) to classify each candidate location and fine-tune the bounding boxes. Gao et al. [7] combined Faster R-CNN and FCN to detect tunnel defects and concurrently perform segmentation for the defects. Adaptive Border RoI boundary layer was also adopted to improve the model performance, which made the false detection rate decrease and identification accuracy improve significantly. In addition, the backbone like ResNet [8], can successfully solve the problem of vanishing/exploding gradients by means of shortcut connections. To recognize objects at different scales, the backbone adopts a feature pyramid network (FPN) [9] to make a compromise between the accuracy of locations and the semantic level of features. Mask R-CNN [10], implemented with complex network structures, such as FPN for extracting image features at different scales and RPN for keeping positive and negative samples balance, is one of the state-of-the-art two-stage networks. As a result, the segmentation performance of the two-stage network outperforms the original FCN.

However, a series of experiments reveal a serious problem in the two-stage algorithm. When the network adopts a series of complicate methods to enhance the detection accuracy rate, the algorithm will be very time-consuming. Because the structural inspection time is usually limited to 2–3 h, a Mask R-CNN with an inference time of 3.24 s per image fails to meet the need of engineering. To deal with this problem, during the stage of model design, a novel convolutional neural network is constructed by means of combining SSD [11] and FCN, hereafter referred to as SSD-FCN. This network achieves a more accurate and faster detection performance than R-CNN through a set of default boxes over different aspects of ratios and scales per feature map location.

The rest of this study is structured as follows. Section 2 presents the datasets with database generation procedure and relevant operation of data augmentation. Section 3 introduces the overall structure of the network. Section 4 gives some model test results and Sect. 5 concludes the article.

2 Datasets

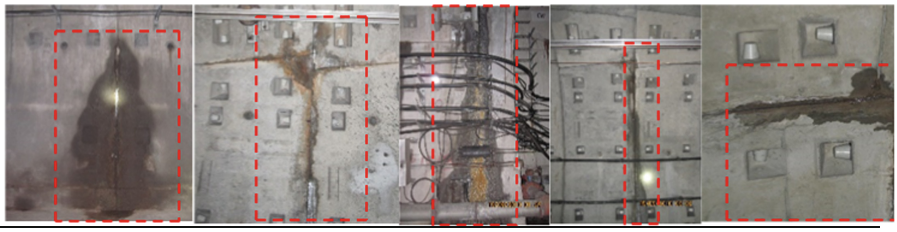
2.1 Data Sources

In the field of deep learning, training a model requires a large dataset. Generally, the higher the data quality, the better the training effect of the model. There are many famous datasets such as ImageNet [12] and Microsoft COCO [13], which perform a crucial role in the rise of deep learning. However, to the best of the authors knowledge, no comprehensive datasets is available containing tunnel defects information. Therefore, it is necessary to collect a sufficient number of tunnel images to achieve an excellent model. To this end, water leakage images of a shield tunnel in Shanghai metro was collected by authors for three months under different tunnel environment and lining defect conditions.

2.2 Datasets Preparing

Given that there are a series of affiliate facilities (e.g., electric cables, pipelines, handholes, etc.) attached to the tunnel lining surface, severe interference should be considered in the recognition of water leakage. Therefore, water leakage is divided into five categories (Table 1) to investigate the segmentation effect in different background conditions. After classification, data augmentation is adopted to keep the balance of different groups.

Table 1. Water leakage images in different backgrounds



1) 2) 3) 4) 5)

1) Leakage and simple background; 2) blocky leakage and handhole or edge joint; 3) leakage and pipelines; 4) vertical strip leakage and complex background; and 5) horizontal strip leakage and complex background (The red regions contain the water leakage of different categories).

2.3 Data Augmentation

Generally, deep learning can work accurately when a comprehensive enough dataset is available. The features extracted automatically also require many positive samples. Therefore, the original datasets containing 711 images in training sets and 200 images in validation sets were enriched by means of data augmentation. Multi-angle image geometrical transformation was adopted to enhance the spatial complexity of the water leakage. The original distribution of different categories of water leakage is shown in Table 2. By means of data augmentation, the datasets were eventually augmented to 3555, five times larger than the original case.

Table 2. Distribution of original datasets

	Training	Validation	Testing	Sum
1)	103	40	13	156
2)	125	40	17	182
3)	247	40	38	325
4)	154	40	22	216
5)	82	40	10	132

3 Network Architecture

The new model contains two stages. In the first stage, SSD detects and crops the leakage region of the original images. Then, the cropped images are sent to the FCN. In the second stage, FCN predicts a segmentation mask in a pixel-to-pixel manner for each water leakage image.

3.1 SSD Architecture

In the field of object detection, the two-stage model (i.e., Mask R-CNN), has achieved high accuracy. However, its two-stages framework, region proposal network and head stage (classification, detection, and segmentation in one branch), decreases the detection speed, and thus, cannot meet the requirement of engineering. As such, SSD removes the network of region proposal and directly uses CNN to classify and regress the sampling area in the image, which not only saves time but also achieves relatively high accuracy. SSD adopts VGG-16 as the backbone and takes some auxiliary measures to optimize the original network structure.

Extra Feature Maps. To solve the problem of prediction over multiple-scale objects, as is shown in Fig. 1, the feature maps computed by lower convolutional layers have weaker semantic but finer information, while the higher convolutional layers obtain stronger semantic but more coarse information. In other words, the feature maps of lower layers can be adopted to detect small targets and the feature maps of higher layers can be adopted to detect big targets.

Bounding Boxes. For each pixel on the feature maps ($W_i \times H_i$) of different levels, bounding boxes with four aspect ratios ($a_r \in$ (1:1, 1:2, 2:1, 1:1)) are generated to detect multiple-scale objects accurately. The scale of the bounding boxes of each feature map can be calculated as Eq. (1). Thus, the height (h_i) and width (w_i) of the bounding boxes are computed as Eq. (2). Notably, there are two kinds of bounding boxes with the same aspect ratio, the scale of the extra bounding box is defined as Eq. (3). In SSD, the feature map of a specific level is utilized to be responsive to particular scales of objects, which indicates that the bounding boxes are not strictly corresponded to the actual receptive fields of the feature maps.

$$s_i = s_{\min} + \frac{s_{\max} - s_{\min}}{n-1}(i-1), \quad i \in (1, n] \tag{1}$$

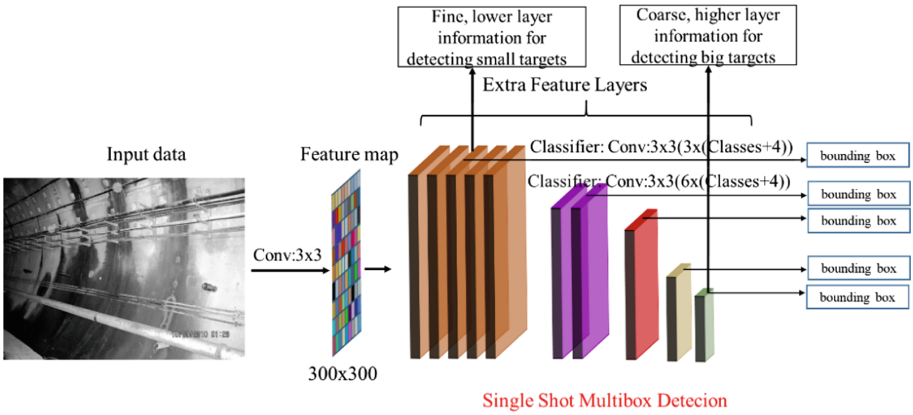


Fig. 1. The network structure of SSD

$$\begin{cases} w_i = W_i \cdot s_i\sqrt{a_r} \\ h_i = H_i \cdot s_i\sqrt{a_r} \end{cases} \tag{2}$$

$$s_i' = \sqrt{s_i s_{i+1}} \tag{3}$$

where $s_{max} = 0.9$ and $s_{min} = 0.2$, indicating the respective scale of the bounding boxes of the lowest and highest feature map. n is the number of feature maps and does not contain the first feature map (4 in this paper).

3.2 FCN Architecture

To achieve high-precision segmentation of an image, FCN uses several convolution layers to replace the fully connected ones in traditional classification models and achieves end-to-end training. In other words, FCN firstly generates a feature map containing advanced semantic features. Then the feature map is up-sampled to the same size as the input image. Finally, the softmax function is adopted to classify up-sampling results pixel by pixel. Moreover, to improve the accuracy and robustness of segmentation, a skip architecture is defined in FCN as shown in Fig. 2. In other words, by means of combining the advantages of the lower and higher feature maps, FCN achieves a relatively high accuracy in the segmentation of water leakage.

3.3 SSD-FCN

As previously explained, SSD and FCN are combined in the proposed algorithm to detect the water leakage accurately and quickly and further achieve a precise segmentation of the water leakage. In the training stage, to make FCN better distinguish the background and the water leakage, the ground truth bounding boxes were enlarged to 1.5 times of the original. While during the testing stage, when bounding boxes were generated by SSD, these boxes were enlarged to 1.2 times of the original and then FCN was utilized to predict the cropped area at the pixel level. The structure of the new model is shown in Fig. 3.

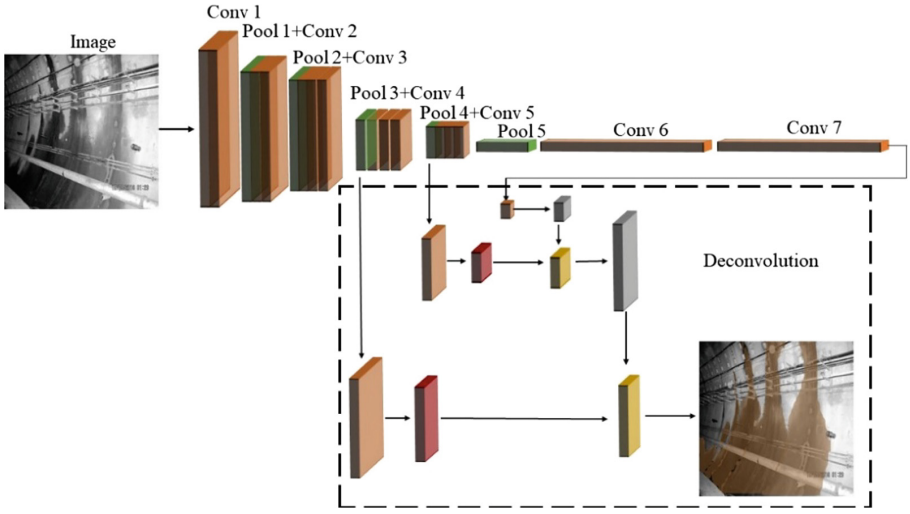


Fig. 2. The network structure of FCN

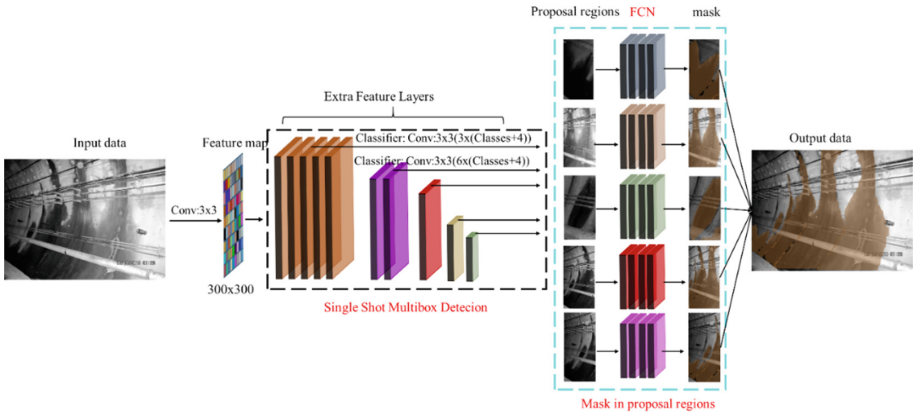
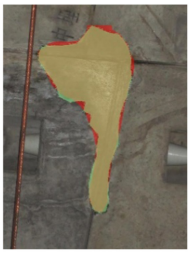
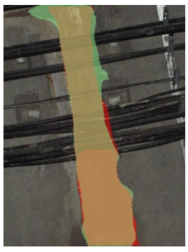
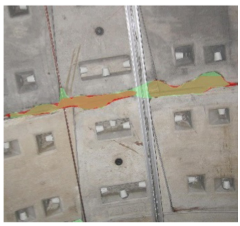


Fig. 3. The structure of the novel model

4 Test Result

A total of 278 images were tested with a speed of 0.09 s per image, while the speed of Mask R-CNN is 3.24 s per image. The precision degree is 70.28%, the recall rate is 66.53%, and the average IoU is 77.25%. Some segmentation results are displayed in Table 3.

Table 3. The test results of the new model (green indicates the ground truth and red represents the prediction)

Category	Predictions and ground results	IoU
1)		0.7500
2)		0.7260
3)		0.7584
4)		0.8480
5)		0.7800

5 Conclusion

The structural inspection of urban tunnel lining is usually expected to be completed within 2–3 h, while it requires a large number of images to be processed. Therefore, in order to process these images efficiently and accurately and further determine the location of the water leakage, this paper proposed a novel convolutional network by combining the advantages of SSD and FCN. The main contributions of this paper are as follows:

1) Combining SSD and FCN, a novel convolutional neural network is constructed in this study which provides an accurate segmentation for the water leakage in the testing stage.
2) In the testing set, SSD-FCN achieves a speed of 0.09 s per image, which is faster than Mask R-CNN, and an average IoU of 77.25% which is close to that of Mask R-CNN.
3) The results indicate that the water leakage with a vertical bar shape and simple background achieves the best segmentation accuracy with an IoU greater than 84.8%. In addition, the remaining four categories of water leakage also obtain similar segmentation accuracy.

Considering that SSD removes the region proposal stage and directly uses a convolutional neural network to detect water leakage, the aspect ratios of the bounding boxes need to be carefully designed to match the water leakage area. Therefore, choosing a set of suitable aspect ratios of the bounding boxes ought to be the focus in future research.

References

1. Xue, Y.D., Li, Y.C.: A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. Comput.-Aided Civ. Infrastr. Eng. **33**(8), 638–654 (2018)
2. Ren, S.Q., He, K.M., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
3. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **4**, 640–651 (2017)
4. Huang, H.W., Li, Q.T.: Image recognition for water leakage in shield tunnel based on deep learning. Chin. J. Rock Mechan. Eng. **36**(12), 2861–2871 (2017)
5. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V.: A review on deep learning techniques applied to semantic segmentation (2017)
6. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intelligence **99**, 2999–3007 (2017)
7. Gao, X., Jian, M., Hu, M., Tanniru, M., Li, S.: Faster multi-defect detection system in shield tunnel using combination of FCN and faster RCNN. Adv. Struct. Eng. **22**(13), 2907–2921 (2019)
8. He, K.M., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 29th IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE, Las Vegas (2016)

9. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 30th IEEE conference on computer vision and pattern recognition, pp. 2117–2125. IEEE, Hawaii (2017)
10. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 1 (2017)
11. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer, Cham (2016)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, F.F.: ImageNet: a large-scale hierarchical image database. In: 22th IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE, Miami (2009)
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L.: Microsoft coco: common objects in context. In: 13th European Conference on Computer Vision, pp. 740–755. Springer, Cham (2014)

Determining the Gain and Directivity of Antennas Using Support Vector Regression

Ezgi Deniz Ulker[1] and Sadık Ulker[2](✉)

[1] Department of Computer Engineering, European University of Lefke, Mersin-10, Gemikonağı, Turkey
eulker@eul.edu.tr

[2] Department of Electrical and Electronics Engineering, European University of Lefke, Mersin-10, Gemikonağı, Turkey
sulker@eul.edu.tr

Abstract. The paper presents the application of support vector regression technique for the modelling of antennas. Two different antennas were modelled with different properties. The first modelling was for a helical antenna with varying parameters to determine the gain of antenna. In the second modelling, an equilateral triangular patch antenna with varying properties for the design was considered for the determination of the directivity of the antenna. The support vector regression modelled both of the antennas very well. In helical antenna, the radial kernel with ν-regression produced only 0.143 dBi average error. In equilateral triangular patch antenna, the radial kernel with ε-regression produced only 0.126 dBi average error.

Keywords: Support vector regression · Kernels · Helical antenna · Equilateral triangular patch antenna

1 Introduction

Different artificial intelligence techniques have found applications in many different variety of problems in engineering and sciences. These took the form of solving problems which are difficult to find using analytical methods, finding parameters as unknowns or optimizing the values of unknown variables in design problems.

Use of support vector machines is one of the most powerful artificial intelligence technique which mainly is applied for classification problems. Similarly, the support vector regression is a very powerful technique applied to many engineering and science problems for prediction. For building energy consumption prediction Zhong et al. used support vector regression recently [1]. An application to financial forecasting was done by Trafalis and Ince [2]. The end effects of Hilbert-Huang transform was studied by Cheng et al. [3]. Ulker suggested the method for the prediction of unemployment rate and GDP [4].

For antenna problems, the application of support vector machines also found great acceptance. Support vector machine was used in antenna selection by Yang et al. [5].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 62–69, 2021.
https://doi.org/10.1007/978-3-030-55180-3_5

Support vector characterization of microstrip rectangular patch antenna was suggested by Tokan and Güneş [6]. For array processing applications, support vector regression formulation was used as an optimization tool [7]. For multiuser communication systems, support vector machine was used for transmit antenna allocation by Lin et al. [8]. Recently, Ulker used support vector regression technique for the design of a microstrip feedline for rectangular patch antenna [9].

The effectiveness of support vector machine in these kind of problems is twofold. With a proper analysis we can obtain a model in which we can quickly and accurately determine a desired design value, and we can predict a desired design value by extrapolating the model.

For antenna modelling support vector regression machines were used by Angiulli et al. [10] for modelling rectangular patch antennas. In this work our aim was to apply the support vector regression for helical antennas and equilateral triangular patch antenna and compare the performance in modelling between using linear and radial kernels and different type of regressions namely ε-regression and ν-regression.

2 Support Vector Machine

The original theory, Vapnik-Chervonenkis theory of statistical learning, dates back to 1960's [11]. Developed in 1990's the main working principle of support vector machine is to construct a hyperplane as the decision surface in a way that the margin of separation between positive and negative examples was maximized [12, 13]. Later a regression technique based on support vector machine was developed [14]. The important feature of support vector regression is that the optimization does not depend on dimensionality of the input space. A descriptive tutorial on support vector regression can be found at Smola et al.'s work [15].

Because of being a very powerful technique, support vector regression has been used in many different applications since its emergence. The method is still developed and application in many different problems in variety of fields can be found in recent years. Prediction of wind speed and direction using artificial intelligence techniques, including support vector regression was performed by Khosravi et al. [16]. Forecasting the high frequency volatility of cryptocurrencies and traditional currencies with support vector regression was suggested by Peng et al. [17]. In a hydrology application, prediction of rainfalls in Iran was done by Mehr et al. [18]. Prediction of river flow in Kızılırmak River in Turkey has been modeled and determined by Baydaroğlu et al. [19].

In support vector regression, a mapping into a high dimensional feature space is necessary. This can be done with the use of various type of kernels, such as linear, polynomial, radial basis, and sigmoid. Also the regression can be done as ε-regression and ν-regression. In ν-regression, we can control the proportion of the number of support vector we keep in the solution with respect to the total number of samples in the dataset. In ε-regression however, we control the parameter ε, which determines how much error we allow the model to have.

In our work, we use radial and linear kernels, also both ε-regression and ν-regression techniques to compare the results produced by each.

3 Results

3.1 Helical Antenna

Helical antennas are antennas consisting of a conducting wire wound in the form of a helix. The radiation occurs along the axis of the antenna in a circularly polarized fashion. The helical antennas are very popular because they can easily be constructed as well as having a large bandwidth of operation.

A simple helical antenna is shown below in Fig. 1. The design parameters are C (circumference of a round), S spacing between the rounds, N number of turns. Circumference of the round is roughly set to the value of one wavelength at the design frequency. The other parameters S and N on the other hand, affect the value of antenna parameter gain.

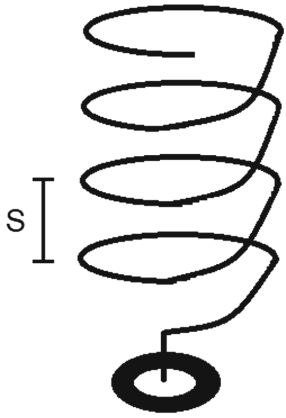


Fig. 1. Simple drawing of a helical antenna.

In order to model helical antenna, calculations were completed to obtain the necessary data. The data were obtained through calculations for various N and S combinations by setting C constant (by fixing the frequency of operation to 1800 MHz). A set of 48 data were obtained. This data was used in support vector regression technique as 45 data for training and 3 for testing. Linear and radial kernels were used with both ε-regression and ν-regression. The response is shown in Fig. 2 and Fig. 3. In both of the figures it can easily be observed that our model created a close approximation to the calculated data.

The comparison of results are tabulated in Table 1.

Table 1. Summary of results for helical antenna.

Kernel	Regression type	Average error
Linear	ν-regression	0.615
Linear	ε-regression	0.617
Radial	ν-regression	0.143
Radial	ε-regression	0.314

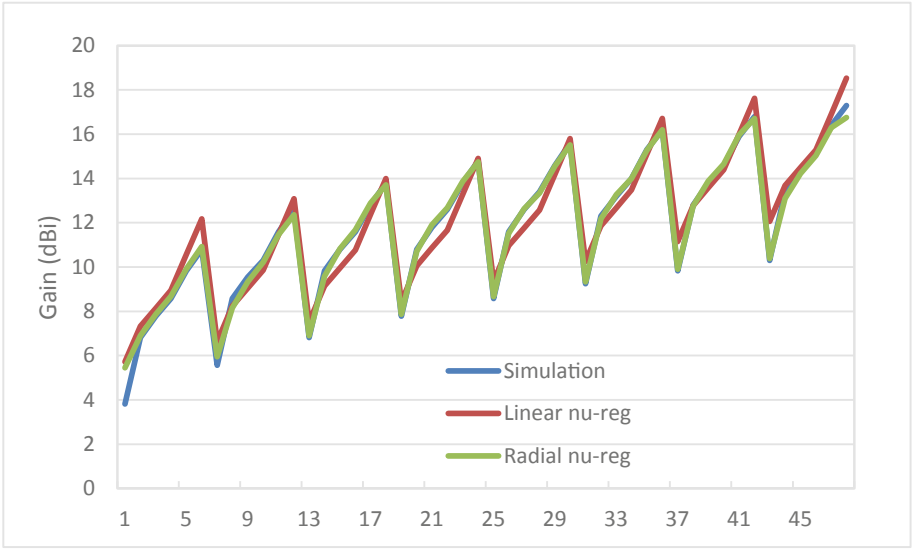


Fig. 2. Helical antenna ν-regression.

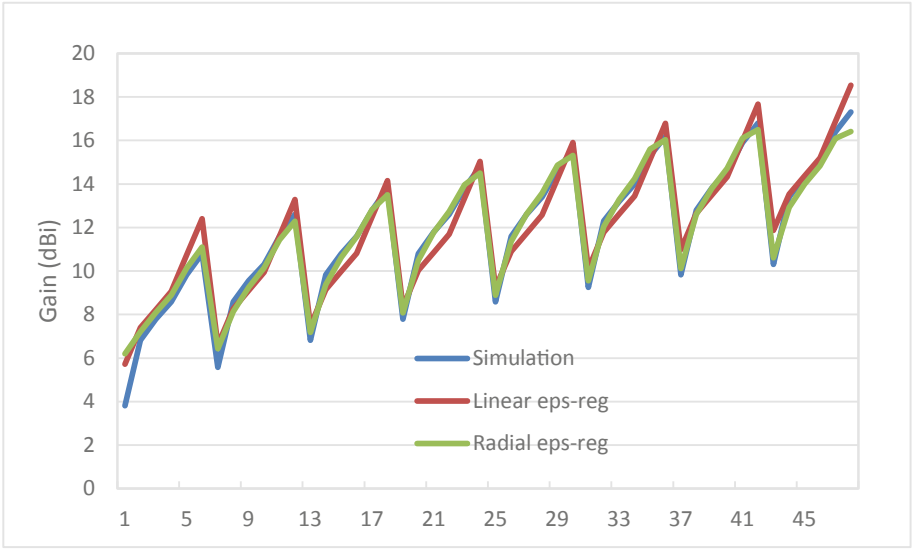


Fig. 3. Helical antenna ε-regression.

3.2 Equilateral Triangular Patch Antenna

Microstrip antennas are very popular because of many attractive features they possess for antenna designers. They are cheap and can be easily fabricated on a dielectric substrate. Mainly used in wireless applications, microstrip antennas can be of many different shapes.

A simple drawing of an equilateral triangular patch antenna on a substrate is shown in Fig. 4. In this work, equilateral triangular patch antenna was simulated with different patch lengths. The simulation result produced resonant frequency (GHz) and directivity values (dBi). In this work, we used the data for patch length (mm) and resonant frequency (GHz) as input values and directivity value (dBi) as output value.

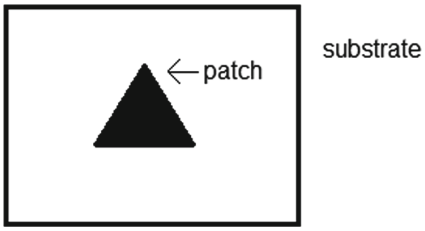


Fig. 4. A drawing (top view) of an equilateral triangular patch antenna.

The data was obtained from the simulation results of Tripathi et al. [20]. Overall, 14 data points were used with 12 for training and 2 for testing. Similarly, linear and radial kernels were used with both ε-regression and ν-regression. The response is shown in Fig. 5 and Fig. 6. In both of the figures, it can clearly be observed that although the predicted results were not as good as the previous case, still the predicted results closely describe the calculated (or simulated) values. The summary of comparison results are tabulated in Table 2.

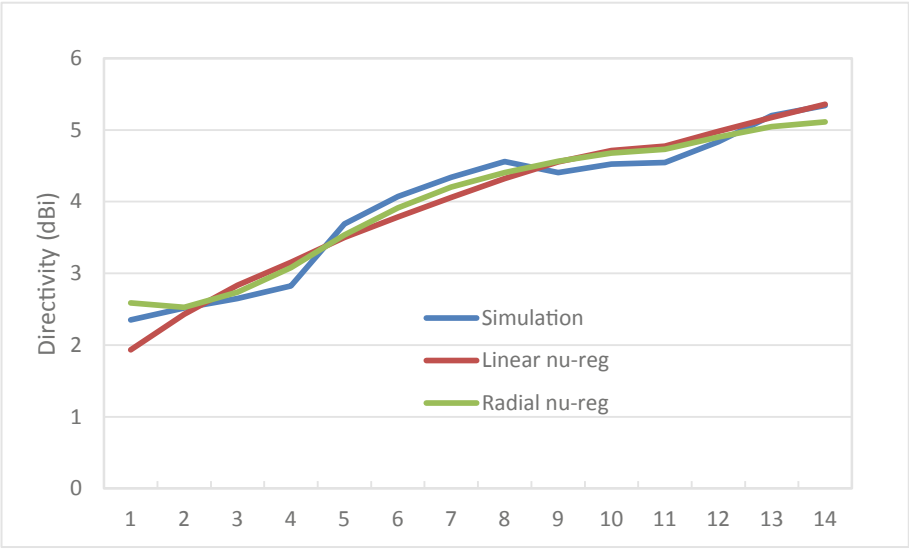


Fig. 5. Equilateral triangular patch antenna ν-regression.

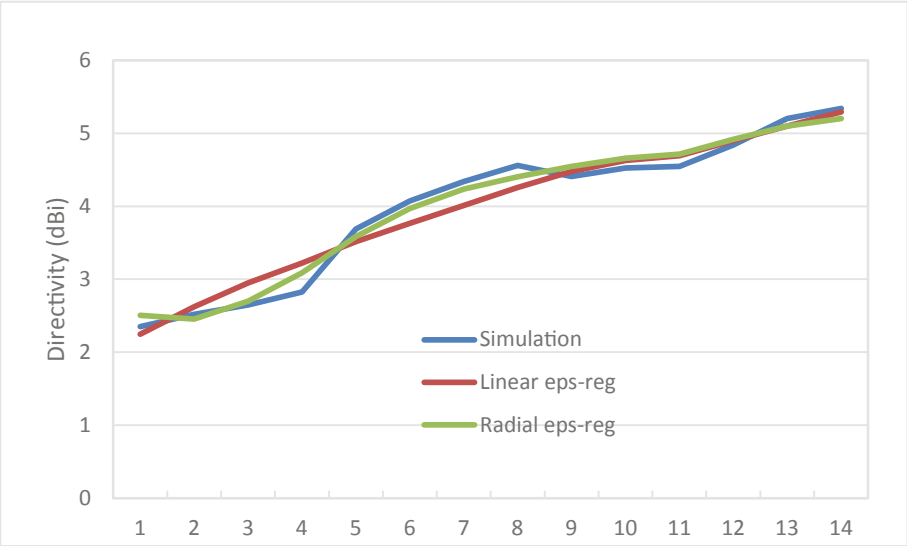


Fig. 6. Equilateral triangular patch antenna ε-regression.

Table 2. Summary of results for equilateral triangular microstrip patch antenna.

Kernel	Regression type	Average error
Linear	ν-regression	0.197
Linear	ε-regression	0.182
Radial	ν-regression	0.152
Radial	ε-regression	0.126

4 Conclusions

The application of support vector regression in modeling two different antennas have been demonstrated. In modelling for gain of helical antenna, it was observed that with radial kernel and ν-regression in average only 0.143 dBi (about 1.2%) difference is observed from the calculated value.

In modelling for directivity of equilateral triangular patch antenna, it was observed that radial kernel with ε-regression gave the best performance. Only 0.126 dBi (about 3.16%) difference is observed from the simulation data.

The calculated values using regression showed the success of support vector regression in modelling. It was observed that the modelling of gain of helical antenna was better when compared with the modelling the directivity of equilateral triangular patch antenna. This is because for the modelling of equilateral triangular patch antenna very few data points were used in training. Hence, with more training data, it is expected to observe better modelling with smaller percentage error values.

References

1. Zhong, H., Wang, J., Jia, H., Mu, Y., Lv, H.: Vector field-based support vector regression for building energy consumption prediction. Appl. Energy **242**, 403–414 (2019). https://doi.org/10.1016/j.apenergy.2019.03.078
2. Trafalis, T.B., Ince, H.: Support vector machine for regression. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (2000). https://doi.org/10.1109/ijjcnn.2000.859420
3. Cheng, J., Yu, D., Yang, Y.: Application of support vector regression machines to the processing of end effects of Hilbert-Huang transform. Mech. Syst. Signal Process. **21**(3), 1197–1211 (2007). https://doi.org/10.1016/j.ymssp.2005
4. Ulker, E.D., Ulker, S.: Unemployment rate and GDP prediction using support vector regression. In: AISS 2019 International Conference on Advanced Information Science and System, Singapore, Article No. 17, pp. 1–5 (2019). https://doi.org/10.1145/3373477.3373494
5. Yang, P., Zhu, J., Xiao, Y., Chen, Z.: Antenna selection for MIMO system based on pattern recognition. Digit. Commun. Netw. **5**, 34–39 (2019). https://doi.org/10.1016/j.dcan.2018.10.0001
6. Tokan, N.T., Güneş, F.: Support vector characterization of the microstrip antennas based on measurements. Prog. Electromagn. Res. B **5**, 49–61 (2008). https://doi.org/10.2528/PIERB08013006
7. Martinez-Ramon, M., Rojo-Alvarez, J.L., Camps-Valls, G., Christodoulou, C.G.: Kernel antenna array processing. IEEE Trans. Antennas Propag. **55**(3), 642–650 (2007). https://doi.org/10.1109/TAP.2007.891550
8. Lin, H., Shin, W.-Y., Joung, J.: Support vector machine-based transmit antenna allocation for multiuser communication systems. Entropy **21**, 471 (2019). https://doi.org/10.3390/e21050471
9. Ulker, S.: Support vector regression analysis for the design of feed in a rectangular patch antenna. In: ISMSIT 3rd International Symposium on Multidisciplinary Innovative Technologies (2019). https://doi.org/10.1109/ismsit.2019.8932929
10. Angiulli, G., Cacciola, M., Versaci, M.: Microwave devices and antenna modelling by support vector regression machines. IEEE Trans. Magn. **43**(4), 1589–1592 (2007). https://doi.org/10.1109/TMAG.2007.892480
11. Vapnik, V.N., Chervonenkis, A.Y.: The uniform convergence of frequencies of the appearance of events to their probabilities. Dokl. Akad. Nauk SSSR **181**(4), 781–783 (1968)
12. Schölkopf, B., Burges, C., Vapnik, V.: Incorporating invariances in support vector learning machines. In: International Conference on Artificial Neural Networks ICANN, vol. 96, pp. 47–52 (1996)
13. Haykin, S.: Neural Networks, 2nd edn. Prentice Hall, New Jersey (1999)
14. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: Mozer, M.C., Jordan, M.I., Petsche, T. (eds.) Advances in Neural Information Processing Systems 9, pp. 155–161. MIT Press, Cambridge, MA (1997)
15. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statist. Comput. **14**(3), 199–222 (2004)
16. Khosravi, A., Koury, R.N.N., Machado, L., Pabon, J.J.G.: Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. Sustain. Energy Technol. Assess. **25**, 146–160 (2018). https://doi.org/10.1016/j.seta.2018.01.001

17. Peng, Y., Albuquerque, P.H.M., de Sa, J.M.C., Padula, A.J.A., Montenegro, M.R.: The best of two worlds: forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. Expert Syst. Appl. **97**, 177–192 (2018). https://doi.org/10.1016/j.eswa.2017.12.004
18. Danandeh Mehr, A., Nourani, V., Karimi Khosrowshahi, V., Ghorbani, M.A.: A hybrid support vector regression–firefly model for monthly rainfall forecasting. Int. J. Environ. Sci. Technol. **16**(1), 335–346 (2018). https://doi.org/10.1007/s13762-018-1674-2
19. Baydaroğlu, Ö., Koçak, K., Duran, K.: River flow prediction using hybrid models of support vector regression with the wavelet transform, singular spectrum analysis and chaotic approach. Meteorol. Atmos. Phys. **130**(3), 349–359 (2017). https://doi.org/10.1007/s00703-017-0518-9
20. Tripathi A., Thakare V.V., Singhal, P.K.: Analysis of different performance parameters of equilateral triangular microstrip patch using artificial neural network. Int. J. Adv. Innov. Thoughts Ideas **2**(3) (2013)

Driving Reinforcement Learning with Models

Meghana Rathi[1(✉)], Pietro Ferraro[2], and Giovanni Russo[3]

[1] IBM, Dublin, Ireland
meghana.rathi@ibm.com
[2] Dyson School of Design Engineering, Imperial College London, South Kensington, London, UK
p.ferraro@imperial.ac.uk
[3] Department of Information and Electronic Engineering and Applied Mathematics, Universita' degli Studi di Salerno, Fisciano, Salerno, Italy
giovarusso@unisa.it
http://sites.google.com/view/giovanni-russo/home

Abstract. In this paper we propose a new approach to complement reinforcement learning (RL) with model-based control (in particular, Model Predictive Control - MPC). We introduce an algorithm, the MPC augmented RL (MPRL) that combines RL and MPC in a novel way so that they can augment each other's strengths. We demonstrate the effectiveness of the MPRL by letting it play against the Atari game Pong. For this task, the results highlight how MPRL is able to outperform both RL and MPC when these are used individually.

Keywords: Model Predictive Control (MPC) · Reinforcement Learning (RL) · Safe and accelerated learning

1 Introduction

Model-free reinforcement learning (RL in what follows) has become a popular paradigm to design autonomous agents [6,19,24,37]. Its key idea is that of learning a policy for a given task by interacting with the environment via a *trial and error* mechanism: essentially, the optimal policy is achieved by exploring the state space and by learning which actions are the best (based on some *reward function*) for a given state. Unfortunately, two key practical disadvantages of RL are its sample inefficiency and its lack of (e.g. safety) guarantees while learning [4,30]. This paper proposes an approach to drive the learning of RL that stems from the following observation: in many applications, such as applications requiring physical interactions between the agent and its environment, while a *full* model of the environment might not be available, at least *parts* of the model, for e.g. a subset of the state space, might be known/identifiable. Motivated by

The work in this paper was completed while in School of Electrical & Electronic Eng. University College Dublin Belfield, Dublin, Ireland.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 70–85, 2021.
https://doi.org/10.1007/978-3-030-55180-3_6

this, we explore how the availability of these *partial* models can be leveraged to *drive*, and indeed accelerate, the learning phase of RL.

The paper is organised as follows. We start with discussing the contributions and give a survey of related works. Then, after giving some background on the key *tools* leveraged in the paper, we introduce the MPRL algorithm, our main contribution. We then outline how the MPRL algorithm was used to learn how to play Pong and present numerical results. We also show how the same algorithm can be applied to a different domain and, in particular, we use MPRL to learn how to control an Inverted Pendulum. Conclusions and future work are finally discussed.

1.1 Contributions

We propose a novel algorithm that complements the capabilities of RL with those of a well known and established model-based control method, Model Predictive Control (MPC). Our algorithm, the MPC augmented RL (MPRL in what follows), combines RL and MPC in a novel way so that they can augment each other's strengths. The MPRL is inspired by the fact that, from a designer's perspective, complex tasks can be often fulfilled by combining together a set of *functionalities* and, for some of these functionalities, either a mathematical model is known or it might be worth devising it. For example, functionalities for which it is worth to *invest* to build a model are these that are critical to satisfy safety requirements: in e.g. an automated driving context, these *critical functionalities* are those directly associated to the prevention of crashes and the braking dynamics of the car can be modeled via differential/difference equations. Given these considerations, the MPRL can be described as follows. At each iteration, it checks whether the environment is in a state for which a mathematical model is available (or can be identified). If this is the case, then the action that the agent will apply is computed by leveraging the model and using MPC to optimize a given cost function. If a model is not available, then MPRL makes use of RL (in particular, we will use Q-Learning) to compute a policy from data. The two counterparts (or components) of the MPRL, i.e. MPC and RL, are interlinked and interact within the algorithm. In particular, MPC both drives the state-space exploration and tunes the RL rewards in order to speed the learning process. To the best of our knowledge, this is a new approach to combine MPC and RL, which is complementary to the recent results on learning MPC [31,32,38]. A further exploration of the use of feedback control to enhance the performance of data-driven algorithms can also be found in [22], which proposes a different mechanism to complement RL with a feedback control loop. In order to illustrate the effectiveness of MPRL, we let it play the *Atari* game *Pong* and also control an inverted pendulum. The results highlight the ability of the algorithm to learn the task, outperforming both RL and MPC when these are used individually. The code of our experiments is available at the repository https://github.com/GIOVRUSSO/Control-Group-Code.

1.2 Related Work

We now briefly survey some related research threads.

Physics Simulation. The idea of using models and simulation environments to develop intelligent reinforcement learning agents has recently been attracting much research attention, see e.g. [17,28]. For example, in [11] it is shown how a physical simulator can be embedded in a deep network, enabling agents to both learn parameters of the environment and improve the control performance of the agent. Essentially, this is done by simulating rigid body dynamics via a linear complementarity problem (LCP) technique [8,10–12,23]. LCP techniques are also used within other simulation environments such as MuJoCo [39], Bullet [21], and DART [18]. Instead, a complementary body of literature investigates the possibility of integrating into networks the mechanisms inspired by the intuitive human ability to understand physics (see e.g. [2,3,7], which leverage the ideas of [34,40]).

Model-Based RL. Although model-free methods, have achieved considerable successes in the recent years, many works suggest that a model-based approach can potentially achieve better performance [1,4,20]. The research in model-based RL is a very active area and it is focused on two main settings. The first one makes use of neural networks and a suitable loss function to simulate the dynamics of interest, whereas another approach makes use of more classical mathematical models closely resembling system identification [29].

Safe RL. The design of safe RL algorithms is a key research topic and different definitions of safety have been proposed in the literature (see e.g. [9,15] and references therein). For example, in [16] the authors relate safety to a set of *error states* associated to dangerous/risky situations while in [36] risk adversion is specified in the reward. A complementary approach is the one of model-based RL where safety is formalized via state space constraints, see e.g. [25]. Examples of this approach include [4,27], where Lyapunov functions are used to show forward invariance of the safety set defined by the constraints. Finally, we note that other approaches include [14], where a priori knowledge of the system is used, in order to craft safe backup policies or [33] in which authors consider uncertain systems and enforce probabilistic guarantees on their performance.

2 Background

We now outline the two building blocks composing MPRL.

Model Predictive Control. MPC is a model-based control technique. Essentially, at each time-step, the algorithm computes a control action by solving an optimization problem having as constraint the dynamics of the system being controlled. In addition to the dynamics, other system requirements (e.g. safety or feasibility requirements) can also be formalized as constraints of the optimization problem [13]. Let $x_k \in \mathbb{R}^n$ be the state variable of the system at time k, $u_k \in \mathbb{R}^m$ be its control input and η_k be some noise. In this letter we consider

discrete-time dynamical systems of the form $x_{k+1} = A_k x_k + D_k + C_k u_k + \eta_k$ with initial condition $x_{initial}$ and where the time-varying matrices have appropriate dimensions. For this system, formally the MPC algorithm generates the control input u_k by solving the problem

$$\begin{aligned} &\underset{x_{0:T}\in\mathcal{X}, u_{0:T}\in\mathcal{A}}{\operatorname{argmin}} \mathbb{E}\left\{\sum_{t=0}^{T} J_t(x_t, u_t)\right\} \\ &\text{s.t.} \quad x_{t+1} = A_t x_t + D_t + C_t u_t + \eta_t, \quad x_0 = x_{initial}, \end{aligned} \tag{1}$$

with $x_{0:T}$ ($u_{0:T}$) denoting the sequence $\{x_0, \ldots, x_T\}$ (resp. $\{u_0, \ldots, u_T\}$) and where: (i) $\mathcal{A}$ and $\mathcal{X}$ are sets modelling the constraints for the valid control actions and states; (ii) $\mathbb{E}\left\{\sum_{t=0}^{T} J_t(x_t, u_t)\right\}$ is the cost function being optimized, i.e. the expected value of $\sum_{t=0}^{T} J_t(x_t, u_t)$; (iii) η_t is a zero-mean white noise with constant and bounded variance. See e.g. [5] for more details.

Q-Learning and Markov Decision Processes. Q-Learning (Q-L) is a model free RL algorithm, whose aim is to find an optimal policy with respect to a finite Markov Decision Process (MDP). We adopt the standard formalism for MDPs. A MDP [35] is a discrete stochastic model defined by a tuple $\langle \mathcal{S}, \mathcal{A}, P, \gamma, \mathcal{R} \rangle$, where: (i) $\mathcal{S}$ is the set of states $s \in \mathcal{S}$; (ii) $\mathcal{A}$ is the set of actions $a \in \mathcal{A}$; (iii) $\mathrm{P}(s'|s, a)$ is the probability of transitioning from state s to state s' under action a; (iv) $\gamma \in [0, 1)$ is the discount factor; (v) $\mathcal{R}(s, a)$ is the reward of choosing the action a in the state s. Upon performing an action, the agent receives the reward $\mathcal{R}(s_t, a_t)$. A policy, π, specifies (for each state) the action that the agent will take and the goal of the agent is that of finding the policy that maximizes the expected discounted total reward. The value $Q^{\pi}(s, a)$, named Q-function, corresponding to the pair (s, a) represents the estimated expected future reward that can be obtained from (s, a) when using policy π. The objective of Q-learning is to estimate the Q-function for the optimal policy π^*, $Q^{\pi^*}(s, a)$. Define the estimate as $Q(s, a)$. The Q-learning algorithm works then as follows: after setting the initial values for the Q-function, at each time step, observe current state s_t and select action a_t, according to policy $\pi(s_t)$. After receiving the reward $R(s_t, a_t)$ update the corresponding value of the Q-function as follows:

$$Q(s_t, a_t) \leftarrow (1 - \alpha) Q(s_t, a_t) + \alpha \left[\mathcal{R}(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a)\right], \tag{2}$$

where $\alpha \in [0, 1]$ is called the learning rate. Notice that the Q-learning algorithm does not specify which policy $\pi(\cdot)$ should be considered. In theory, to converge the agent should try every possible action for every possible state many times. For practical reasons, a popular choice for the Q-learning policy is the ϵ-greedy policy, which selects its highest valued (greedy) action, $\pi_\epsilon(s_t) = \operatorname{argmax}_{a_t} Q(s_t, a_t)$, with probability $1 - \epsilon(k-1)/k$ and randomly selects among all other k actions with probability ϵ/k [41].

3 The MPRL Algorithm

We are now ready to introduce the MPRL algorithm, the key steps of which are summarized as pseudo-code in Algorithm 1. The main intuition behind this algorithm is that, in many real world systems, tasks can be broken down into a set of functionalities and, for some of these, a mathematical model might be available. Given this set-up, the MPRL aims at combining the strengths of MPC and Q-L. Indeed: (i) within MPRL, MPC can directly control the agent whenever a model is available and, at the same time, it drives the state exploration of Q-L and adjusts its rewards; (ii) on the other hand, Q-L generates actions whenever no mathematical model is available and hence classic model-based control algorithm could not be used.

The algorithm takes as input the following design parameters: (i) the set of allowed actions, $\mathcal{A}$; (ii) the time horizon and cost function used in (1); (iii) the constants $\overline{r}$ and $\underline{r}$, used by MPC to fine tune the reward of Q-L; (iv) an initial matrix $Q(s, a)$. Then, following Algorithm 1 the following steps are performed:

S1: at each time-step, MPRL checks whether a model is available. As we will see in Sect. 4, for the Pong game, a model can be identified when MPRL is defending against attacks. Instead, in Sect. 5, for an inverted pendulum, we use a predefined action around certain operating conditions of the pendulum arm;

S2a: if a model is available, then the action applied by the agent is generated via MPC. Even if the action applied by the agent is given by MPC, the action that would have been obtained via Q-L is also computed. This is done to enable MPC to drive RL. Indeed, if the action from MPC and Q-L are the same, then the $Q(s, a)$ matrix is updated by using the positive reward $\overline{r}$. On the other hand, if the actions from MPC and Q-L differ from one another, then the $Q(s, a)$ matrix is updated with a non-positive reward $\underline{r}$;

S2b: if a model is not available (or cannot be identified), then the agent's action is generated by Q-L;

S3: all relevant quantities are saved within the main algorithm loop.

4 Using MPRL to Learn Pong

We now illustrate the effectiveness of MPRL by letting it play against the Atari game *Pong*. Pong is a 2 player game where each player moves a paddle in order to bounce a ball to the opponent. A player scores a point when the opponent fails to bounce the ball back and the game ends when a player scores 21 points. In what follows, we give a thorough description of how Algorithm 1 has been implemented in order to allow MPRL to play against Pong.

Algorithm 1. MPRL Algorithm

Inputs:
Allowed actions, $\mathcal{A}$
Time horizon, T, and cost function $\sum_{t=0}^{T} J_t(x_t, u_t)$
Constants $\overline{r}$ and $\underline{r}$
Initial matrix $Q(s, a)$
Main loop:
for $k = 0, \ldots$ **do**
    Check if the model in (1) is known or can be identified
    if Model is available **then**
        Get s_k and x_k
        if $k \geq 1$ **then**
            if $a_{k-1} = u_{k-1}$ **then**
                $Q(s_{k-1}, u_{k-1}) \leftarrow (1-\alpha)Q(s_{k-1}, u_{k-1}) + \alpha\left[\overline{r} + \gamma \max_a\{Q(s_k, a)\}\right]$
            else
                $Q(s_{k-1}, u_{k-1}) \leftarrow (1-\alpha)Q(s_{k-1}, u_{k-1}) + \alpha\left[\underline{r} + \gamma \max_a\{Q(s_k, a)\}\right]$
            end if
        end if
        $x_{init} \leftarrow x_k$ in (1)
        Compute u_k via MPC and a_k using Q-L
        Apply u_k
        Save x_k, s_k, a_k, u_k
    else
        Apply a_k computed via Q-L
        Save s_k, a_k
    end if
    $k \leftarrow k + 1$
end for

4.1 The Environment and Data Gathering

The environment of the game was set-up using the OpenAI gym library in Python. In particular, we used the *PongDeterministic-v4* (with 4 frame skips) configuration of the environment, which is the one used to assess Deep Q-Networks, see e.g. [26]. The configuration used has, as observation space, Box $(210, 160, 3)$ (see Fig. 1, left panel)[1]. Within our experiments, we first removed the part of the images that contained the game score and this yielded an observation space of Box $(160, 160, 3)$ and then we down-sampled the resulting image to get a *reduced* observation space of Box $(80, 80, 3)$ so that each frame consists of a matrix of 80×80 pixels. Within the image, a coordinate system is defined within the environment, with the origin of the x and y axes being in the bottom-right corner.

Given the above observation space, both the position of the ball and the vertical position of the paddle moved by MPRL were extracted from each frame. In particular:

[1] See http://gym.openai.com/docs/ for documentation on the environment observation space.

– At the beginning of the game, the centroid of the ball is found by iterating through the frame to find the location of all pixels with a value of 236 (this corresponds to the white color, i.e. the color of the ball in Pong). Then, once the ball is found the first time, the frame is only scanned in a window around the position of the ball previously found (namely, we used a window of 80×12 pixels, see Fig. 1, right panel);
– Similarly the paddle's centroid position is found by scanning the frame for pixels having value 92 (this corresponds to the green color, i.e. the color of the MPRL paddle).

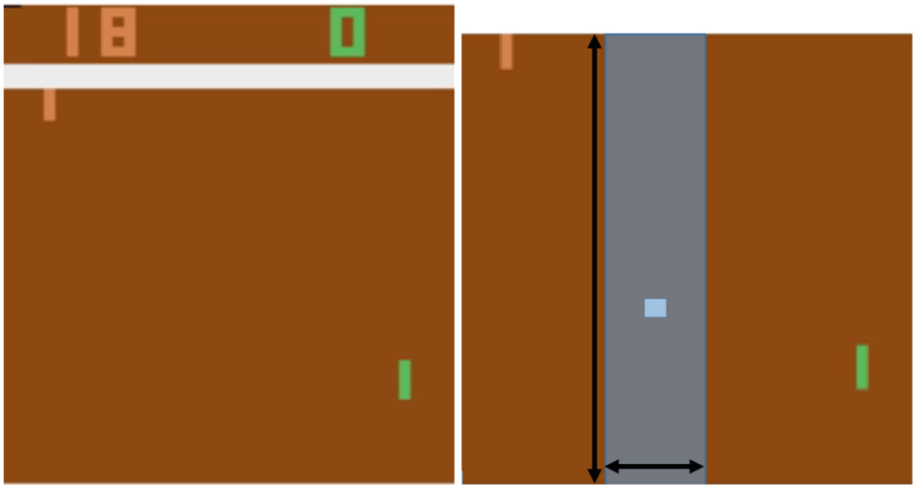


Fig. 1. Left panel: a typical frame from Pong. The paddle moved by MPRL is the green one. Right panel: a zoom illustrating the 80×12 pixels window used to extract the new position of the ball, given its previous position.

4.2 Definition of the Task and Its Functionalities

The agent's task is that of winning the game, which essentially consists of two phases: (i) *defense* phase, where the agent needs to move the paddle to bounce the ball in order to avoid that the opponent makes a point; (ii) an *attack* phase, where the agent needs instead to properly bounce the ball in order to make the point. During the defense phase, MPRL used its MPC component (implemented as described in Sect. 4.3) to move the paddle. Indeed, this phase is completely governed by the *physics* of the game and by the moves of our agent. This, in turn, makes it possible to identify, from pixels, both the ball and the paddle dynamics (Sect. 4.3). Instead, we used the Q-L component of MPRL during the attack phase. Indeed, even if a mathematical model describing the evolution of the

position of the ball could be devised, there is no difference equation that could predict what our opponent would do in response (as we have no control over it). Therefore, in the attack phase, we let the Q-L counterpart of our algorithm learn how to score a point.

4.3 Implementing MPC

We describe the MPC implementation used within MPRL to play Pong by first introducing the mathematical model serving as the constraint in (1). This model describes both the dynamics of the ball and of the paddle moved by MPRL.

Ball Dynamics. We denote by $x_t^{(b)}$ and $y_t^{(b)}$ the x and y coordinates of the centroid of the ball at time t. The mathematical model describing the dynamics of the ball is then:

$$x_{t+1}^{(b)} = x_t^{(b)} + v_{t,x}, \qquad y_{t+1}^{(b)} = y_t^{(b)} + v_{t,y}, \tag{3}$$

where $x_{t+1}^{(b)}$ and $y_{t+1}^{(b)}$ are the predicted next coordinates at time $t+1$ and where the speeds at time t, i.e. $v_{t,x}$ and $v_{t,y}$ are computed from the positions extracted from the current and the two previous frames, i.e. $x_{t-2}^{(b)}, x_{t-1}^{(b)}, x_t^{(b)}$ and $y_{t-2}^{(b)}, y_{t-1}^{(b)}, y_t^{(b)}$. In particular, this is done by first computing the quantities v_{x1}, v_{x2} and v_{y1}, v_{y2} as follows:

$$\begin{bmatrix} v_{x_1} \\ v_{y_1} \end{bmatrix} = \begin{bmatrix} x_t^{(b)} \\ y_t^{(b)} \end{bmatrix} - \begin{bmatrix} x_{t-1}^{(b)} \\ y_{t-1}^{(b)} \end{bmatrix}, \quad \begin{bmatrix} v_{x_2} \\ v_{y_2} \end{bmatrix} = \begin{bmatrix} x_{t-1}^{(b)} \\ y_{t-1}^{(b)} \end{bmatrix} - \begin{bmatrix} x_{t-2}^{(b)} \\ y_{t-2}^{(b)} \end{bmatrix}. \tag{4}$$

Consider now the speed along the x axis. If there is no impact between the ball and the paddle, we set $v_{t,x} = 0.5(v_{x1} + v_{x2}) + var\left([v_{x1}, v_{x2}]\right) = \bar{v}_{t,x} + \eta_{t,x}$ (where $var(a)$ denotes the variance of the generic vector a and $\eta_{t,x}$ is a white noise with zero mean and variance $var\left([v_{x1}, v_{x2}]\right)$). Instead, along the y axis, we have $v_{t,y} = 0.5(v_{y1} + v_{y2}) + var\left([v_{y1}, v_{y2}]\right) = \bar{v}_{t,x} + \eta_{t,x}$ (with $\eta_{t,y}$ being a white noise with zero mean and variance $var\left([v_{y1}, v_{y2}]\right)$) if there has been no impact and $v_{t,x} = v_{x1}$ if there has been an impact of the ball with one of the walls.

Paddle Dynamics. In the gym environment used for the experiments, the only control action that can be applied by MPRL at time t, i.e. u_t is that of moving its paddle. In particular, the agent can either move the paddle up ($u_t = 1$) or down ($u_t = -1$) or simply not moving the paddle ($u_t = 0$). It follows that given the vertical position of the centroid of the paddle at time t, say $y_t^{(p)}$, its dynamics can be modeled by

$$y_{t+1}^{(p)} = y_t^{(p)} + u_t. \tag{5}$$

The MPC Model and the Cost Function. Combining the models in (3)–(5) yields the dynamical system serving as constraint in (1). Note that the resulting model can be formally written as the system in (1) once the state x_t is defined as $x_t = [x_t^{(b)}, y_t^{(b)}, y_t^{(p)}]^T$. Finally, in the implementation of the MPC algorithm,

we used as cost function $\sum_{t=0}^{T} J_t = \|y_{t+T}^{(b)} - y_{t+T}^{(p)}\|^2$. That is, with this choice of cost function the algorithm seeks to regulate the paddle's position so that the distance between the position of the ball at time t and the position of the MPRL paddle at time $t + T$ is minimised. Note that the time horizon, T, used in the above cost function is obtained by propagating the ball model (3) in order to estimate after how many iterates the ball will hit the border protected by the MPRL paddle.

4.4 Implementing Q-L

We implemented the Q-L algorithm outlined in Sect. 4. The set of actions available to the agent were $a_t \in \{-1, 0, +1\}$, while the state at time s_t was defined as the 5-dimensional vector containing: (i) the coordinates of the position at time time t of the ball (in pixels); (ii) the velocity of the ball (rounded to the closest integer) across the x and y axes; (iii) the position of the paddle moved by MPRL. The reward was obtained from the game environment: our agent was given a reward of $+1$, each time the opponent missed to hit the ball, and -1 each time our agent missed to hit the ball. Finally, the values of the Q-table were initialized to 0. In the experiments, the state-action pair was updated whenever a point was scored and a greedy policy was used to select the action. Also, in the experiments we set both α and γ in (2) to 0.7. Moreover, following Algorithm 1, the Q-function was also updated when our agent was defending (i.e. when MPRL was using MPC to move the paddle). In particular, within the experiments we assigned: (i) a positive reward, $\overline{r}$, whenever the action from Q-L and MPC were the same; (ii) a non-positive reward, $\underline{r}$, whenever the actions were not the same. In this way, within MPRL, the Q-L component is driven to learn defence tactics too.

4.5 Handover Between MPRL Components

Finally, we now describe how the handover between the MPC and RL components of MPRL was implemented in the experiments. Intuitively, the paddle was moved by the MPC component when the ball was coming towards the MPRL paddle and, at the same time, the future vertical position of the ball (predicted via the model) was far from the actual position of the agent's paddle. That is, MPC was used whenever the following conditions were simultaneously satisfied: $v_{t,x}^{(b)} < 0$, $\|y_{t+T}^{(b)} - y_t^{(p)}\| > H_y$. In all the other situations the paddle was moved by actions generated by the Q-L component. Note that: (i) $v_t^{(b)}$ is estimated from the game frames as described above (in the environment negative velocities along the x axis mean that the ball is coming towards the green paddle); (ii) the computation of $y_{t+T}^{(b)}$ relies on simulating the model describing the ball's dynamics (3); (iii) H_y is a threshold and this is a design parameter.

4.6 Results

We are now ready to present the results obtained by letting MPRL play Pong. The results are quantified by plotting the *game reward* as a function of the number of *episodes* played by MPRL. An episode consists of as many rounds of pong it takes for one of the players to reach 21 points, while the game reward is defined to be the difference between the points scored by MPRL within the episode and the points scored by the opponent within the episode. Essentially, a negative game reward means that MPRL lost that episode; the lowest possible value that can be attained is -21 and this happens when MPRL is not able to score any point. Viceversa, a positive game reward means that the agent was able to beat the opponent; the maximum value that can be obtained is $+21$, when the opponent did not score any point.

As a first experiment, we implemented an agent that would only use MPC or the Q-L algorithm (without prior training). We let this agent play Pong for 50 episodes and, as the left panel in Fig. 2 shows, as expected, the Q-L agent did not obtain good rewards in the first 50 episodes, consistently loosing games with a difference in the scores of about 20 points. Instead, when the agent used the MPC described in Sect. 4 better performance were obtained. These performance, however, were not comparable with those obtained via a trained Q-L agent and the reason for this is that, while MPC allows the agent to defend, it does not allow for the learning of an attack strategy to consistently obtain points. Using MPRL allowed to overcome the shortcomings of the MPC and Q-L agents. In particular, as shown in the right panel of Fig. 2, when the MPRL agent played against Pong, it was able to consistently beat the game (note the agent never lost a game) while quickly learning an attack strategy to obtain high game rewards. Indeed, note how the agent is able to consistently obtain rewards of about 20 within 50 episodes.

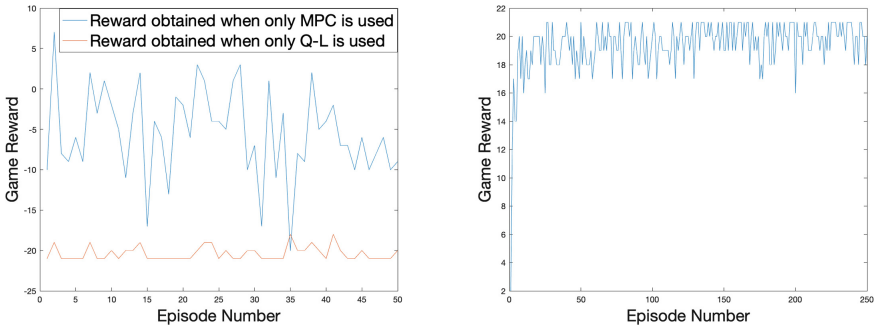


Fig. 2. Left panel: episode rewards as a function of the number of episodes played when either MPC or Q-L are used. Right panel: episode rewards as a function of the number of episodes played by MPRL. Parameters of MPRL were set as follows: $\overline{r} = 0.1$, $\underline{r} = 0$, $H_y = 5$.

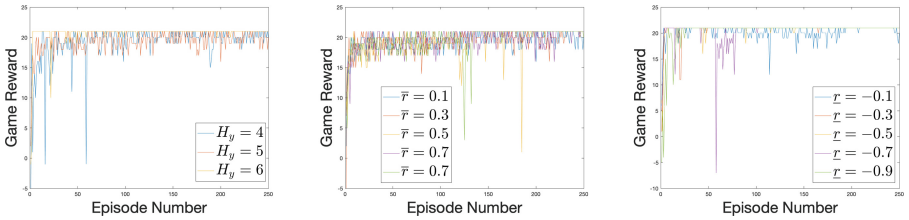


Fig. 3. Left panel: rewards obtained by the MPRL when H_y is perturbed. All the other parameters were kept unchanged (i.e. $\overline{r} = 0.1$, $\underline{r} = 0$). Middle panel: rewards obtained by MPRL when $\overline{r}$ is perturbed. In this experiment, $H_y = 5$ and $\underline{r} = 0$. Right panel: rewards obtained by MPRL when $\underline{r}$ is perturbed. In the experiment, $H_y = 5$ and $\overline{r} = 0.1$.

In order to further investigate the performance of MPRL, we also evaluate its performance when the parameters $\overline{r}$, $\underline{r}$ and H_y are changed. We first take H_y in consideration; H_y determines when the handover between MPC and RL takes place. As Fig. 3 (left panel) illustrates, MPRL is still able to consistently obtain high rewards when H_y is perturbed, i.e. $H_y \in \{4, 5, 6\}$. Notice that when $H_y = 4$, while MPRL is still able to beat the game, it also experiences some drops in the rewards. This phenomenon, which will be further investigated in future studies, might be due to the fact that restricting the space of movements of the Q-L component of the algorithm also restricts its learning capabilities. Instead, when the manoeuvre space of the Q-L is bigger ($H_y = 6$), the algorithm, due to the increased flexibility in the moves, is able to learn better moves faster. As a further experiment, we fixed $H_y = 5$ and perturbed the algorithm parameter $\overline{r}$ so that $\overline{r} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The results of this experiment are shown in the middle panel of Fig. 3. It can be noted that smaller values of $\overline{r}$ (i.e. $\overline{r} = 0.1$ and $\overline{r} = 0.3$) lead to a more consistent performance as compared to the higher values of $\overline{r}$ which lead to negative spikes in the performance. This behaviour might be due to the fact that higher values of $\overline{r}$ essentially imply that MPRL trusts more the MPC actions than Q-L actions, hence penalizing the ability of MPRL to quickly learn the attack strategy. Intuitively, simulations show that, while the MPC component is important for enhancing the agent's defense, *too much influence* of this component on the Q-L can reduce the attack performance of the agent (this, in turn, is essential in order to score higher points). Consistently, the same behaviour can be observed when $\underline{r}$ is perturbed ($H_y = 5$, $\overline{r} = 0.1$ and $\underline{r} \in \{-0.1, -0.3, -0.5, -0.7, -0.9\}$), as shown in Fig. 3 (right panel), where it can be observed that the most negative values $\underline{r}$ lead to dips in performance.

5 MPRL to Control an Inverted Pendulum

An additional experiment was carried out to test the capability of the MPRL algorithm in a different domain. This experiment involved balancing an inverted pendulum in a virtual environment using MPRL and comparing its results

against using Q-L only. Specifically, the goal of the MPRL was to move the cart by controlling its speed so that the pendulum would stay in its upright position at 180° (see Fig. 4).

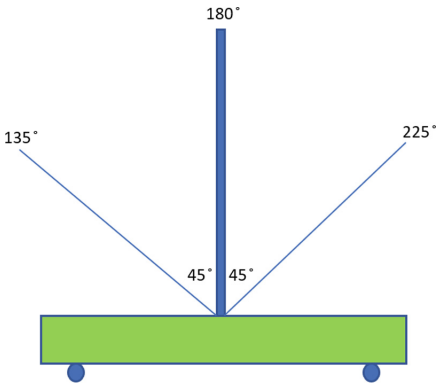


Fig. 4. Inverted pendulum in its balanced state within the safety limits at 135 and 225°.

In our experiments, the reward was specified to be +1 if the pendulum was within 175 and 185° for each time step and −1 if the pendulum went outside this range. Moreover, in our experiments we considered the region between 135 and 225° as a *safe* region and we wanted the MPRL to keep the arm of the pendulum within this region, see Fig. 5. In order to handle this *safety* requirement, in our MPRL we used an MPC-like algorithm whenever the pendulum arm was outside the region 135–225°. In particular: (i) if the arm position was at 135° (or smaller), then the cart speed was adjusted by MPRL to move the cart to the left; (ii) if the arm position was at 225° or higher, then the cart was moved by MPRL to the right. By doing so, the numerical results showed that *MPC component* of MPRL was able to bring the arm within the safety region (i.e. between 135 and 225°). Inside this region, the Q-L component of MPRL was active.

Again, as shown in Fig. 5, the results confirm the capability of the MPRL to quickly learn how to balance the pendulum. Moreover, when compared to the results obtained via Q-Learning, the MPRL dramatically reduces the violation of the *safety* constraints (i.e. the number of times when the pendulum is outside the safe region) and learns faster. This again shows that MPC, and more generally models, can augment the performance of purely data-driven techniques.

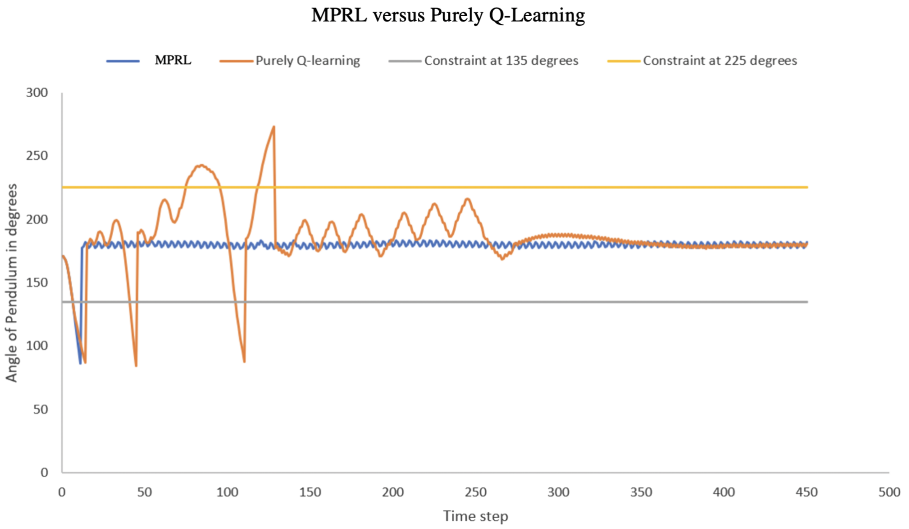


Fig. 5. Angle of the inverted pendulum against time when MPRL is used and when Q-learning is used. The upright position is 180° and the constraints at 135 and 225° are the angles past which the pendulum is highly likely to fall over.

6 Conclusions and Future Work

We investigated the possibility of combining Q-L and model-based control so that they can augment each other's capabilities. In doing so, we introduced a novel algorithm, the MPRL that: (i) leverages MPC when a mathematical model is available and Q-L otherwise; (ii) uses MPC to both drive the state-space exploration of Q-L and to fine tune its rewards. We first illustrated the effectiveness of our algorithm by letting it play against Pong and by analysing its performance when the algorithm parameters are perturbed. Interestingly, the experiments highlight how the algorithm is able to outperform both Q-L and MPC when these are used individually. Moreover, we also tested MPRL in a different domain and showed how it quickly learns to balance an inverted pendulum, while reducing the number of violations of safety constraints. Our future work will include the implementation of the MPRL in different settings, the study of its convergence properties and the design of optimal handover strategies.

References

1. Atkeson, C.G., Santamaria, J.C.: A comparison of direct and model-based reinforcement learning. In: International Conference on Robotics and Automation, pp. 3557–3564 (1997)
2. Battaglia, P., Pascanu, R., Lai, M., Rezende, D.J., Kavukcuoglu, K.: Interaction networks for learning about objects, relations and physics. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 4509–4517 (2016)

3. Battaglia, P.W., Hamrick, J.B., Tenenbaum, J.B.: Simulation as an engine of physical scene understanding. Proc. Natl. Acad. Sci. **110**(45), 18327–18332 (2013)
4. Berkenkamp, F., Turchetta, M., Schoellig, A., Krause, A.: Safe model-based reinforcement learning with stability guarantees. In: Advances in Neural Information Processing Systems, vol. 30, pp. 908–918 (2017)
5. Borrelli, F., Bemporad, A., Morari, M.: Predictive Control for Linear and Hybrid Systems, 1st edn. Cambridge University Press, New York (2017)
6. Breyer, M., Furrer, F., Novkovic, T., Siegwart, R., Nieto, J.: Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning. IEEE Robot. Autom. Lett. **4**(2), 1549–1556 (2019)
7. Chang, M.B., Ullman, T., Torralba, A., Tenenbaum, J.B.: A compositional object-based approach to learning physical dynamics. In: 5th International Conference on Learning Representations (2017)
8. Cline, M.B.: Rigid body simulation with contact and constraints. Ph.D. thesis (2002)
9. Coraluppi, S.P., Marcus, S.I.: Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes. Automatica **35**(2), 301–309 (1999)
10. Cottle, R.W.: Linear complementarity problem. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 1873–1878. Springer, Boston (2009)
11. de Avila Belbute-Peres, F., Smith, K., Allen, K., Tenenbaum, J., Kolter, J.Z.: End-to-end differentiable physics for learning and control. In: Advances in Neural Information Processing Systems, vol. 31, pp. 7178–7189 (2018)
12. Degrave, J., Hermans, M., Dambre, J., Wyffels, F.: A differentiable physics engine for deep learning in robotics. Front. Neurorobot. **13**, 6 (2019)
13. García, C.E., Prett, D.M., Morari, M.: Model predictive control: theory and practice–a survey. Automatica **25**(3), 335–348 (1989)
14. García, J., Fernández, F.: Safe exploration of state and action spaces in reinforcement learning. J. Artif. Int. Res. **45**, 515–564 (2012)
15. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. **16**, 1437–1480 (2015)
16. Geibel, P., Wysotzki, F.: Risk-sensitive reinforcement learning applied to control under constraints. J. Artif. Int. Res. **24**, 81–108 (2005)
17. Hazara, M., Kyrki, V.: Transferring generalizable motor primitives from simulation to real world. IEEE Robot. Autom. Lett. **4**(2), 2172–2179 (2019)
18. Hermans, M., Schrauwen, B., Bienstman, P., Dambre, J.: Automated design of complex dynamic systems. PLOS One **9**(1), 1–11 (2014)
19. Hoppe, S., Lou, Z., Hennes, D., Toussaint, M.: Planning approximate exploration trajectories for model-free reinforcement learning in contact-rich manipulation. IEEE Robot. Autom. Lett. **4**(4), 4042–4047 (2019)
20. Kurutach, T., Clavera, I., Duan, Y., Tamar, A., Abbeel, P.: Model-ensemble trust-region policy optimization. In: International Conference on Learning Representations (2018)
21. Lee, J., Grey, M.X., Ha, S., Kunz, T., Jain, S., Ye, Y., Srinivasa, S.S., Stilman, M., Liu, C.K.: DART: dynamic animation and robotics toolkit. J. Open Sour. Softw. **3**(22), 500 (2018)
22. De Lellis, F., Auletta, F., Russo, G., di Bernardo, M.: Control-tutored reinforcement learning: an application to the herding problem (2019)
23. Lerer, A., Gross, S., Fergus, R.: Learning physical intuition of block towers by example. In: 33rd International Conference on Machine Learning, vol. 48, pp. 430–438 (2016)

24. Liu, B., Wang, L., Liu, M.: Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. IEEE Robot. Autom. Lett. **4**(4), 4555–4562 (2019)
25. McKinnon, C.D., Schoellig, A.P.: Learn fast, forget slow: safe predictive learning control for systems with unknown and changing dynamics performing repetitive tasks. IEEE Robot. Autom. Lett. **4**(2), 2180–2187 (2019)
26. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. In: NIPS Deep Learning Workshop (2013)
27. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
28. Pecka, M., Zimmermann, K., Petrlík, M., Svoboda, T.: Data-driven policy transfer with imprecise perception simulation. IEEE Robot. Autom. Lett. **3**(4), 3916–3921 (2018)
29. Pecka, M., Svoboda, T.: Safe exploration techniques for reinforcement learning – an overview. In: Hodicky, J. (ed.) Modelling and Simulation for Autonomous Systems, pp. 357–375 (2014)
30. Pfeiffer, M., Shukla, S., Turchetta, M., Cadena, C., Krause, A., Siegwart, R., Nieto, J.: Reinforced imitation: sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations. IEEE Robot. Autom. Lett. **3**(4), 4423–4430 (2018)
31. Rosolia, U., Borrelli, F.: Learning model predictive control for iterative tasks. A data-driven control framework. IEEE Trans. Autom. Control **63**(7), 1883–1896 (2018)
32. Rosolia, U., Zhang, X., Borrelli, F.: Data-driven predictive control for autonomous systems. Ann. Rev. Control Robot. Auton. Syst. **1**(1), 259–286 (2018)
33. Sadigh, D., Kapoor, A.: Safe control under uncertainty with probabilistic signal temporal logic. In: Robotics: Science and Systems XII (2016)
34. Smith, K.A., Vul, E.: Sources of uncertainty in intuitive physics. Top. Cogn. Sci. **5**(1), 185–199 (2013)
35. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning, 1st edn. MIT Press, Cambridge (1998)
36. Tamar, A., Mannor, S., Xu, H.: Scaling up robust MDPS using function approximation. In: Proceedings of the 31st International Conference on Machine Learning, vol. 32, pp. 181–189 (2014)
37. Tan, X., Chng, C., Su, Y., Lim, K., Chui, C.: Robot-assisted training in laparoscopy using deep reinforcement learning. IEEE Robot. Autom. Lett. **4**(2), 485–492 (2019)
38. Thananjeyan, B., Balakrishna, A., Rosolia, U., Li, F., McAllister, R., Gonzalez, J.E., Levine, S., Borrelli, F., Goldberg, K.: Safety augmented value estimation from demonstrations (saved): safe deep model-based RL for sparse cost robotic tasks (2019)
39. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033, October 2012

40. Werbos, P.J.: Neural networks for control and system identification. In: Proceedings of the 28th IEEE Conference on Decision and Control, vol. 1, pp. 260–265, December 1989
41. Wunder, M., Littman, M.L., Babes, M.: Classes of multiagent Q-learning dynamics with epsilon-greedy exploration. In: Proceedings of the 27th International Conference on Machine Learning, pp. 1167–1174 (2010)

Learning Actions with Symbolic Literals and Continuous Effects for a Waypoint Navigation Simulation

Morgan Fine-Morris[1], Bryan Auslander[2], Hector Muños-Avila[1(✉)], and Kalyan Gupta[2]

[1] Lehigh University, Bethlehem, PA 18015, USA
hem4@lehigh.edu
[2] Knexus Research Corporation, National Harbor, MD 20745, USA

Abstract. We present an algorithm for learning planning actions for waypoint simulations, a crucial subtask for robotics, gaming, and transportation agents that must perform locomotion behavior. Our algorithm is capable of learning operator's symbolic literals and continuous effects even under noisy training data. It accepts as input a set of preprocessed positive and negative simulation-generated examples. It identifies symbolic preconditions using a MAX-SAT constraint solver and learns numeric preconditions and effects as continuous functions of numeric state variables by fitting a logistic regression model. We test the correctness of the learned operators by solving test problems and running the resulting plans on the simulator.

Keywords: Learning action models · MAX-SAT · Logistic regression

1 Introduction

A recurring problem in applying automated planning technology to practical problems is the need to manually encode domains with a collection of actions that generalize all possible actions. Automated learning of such operators is a possible solution to the problem of manual operator encoding and has been a recurrent research topic as we will discuss in the related work section.

In this paper, we are interested in the automated learning of planning actions for waypoint simulations. Waypoint simulations are a crucial subtask for mobile networks [1], gaming agents [2], and robotics agents [3] that must perform locomotion behavior. Learning operators in this domain requires reasoning with continuous-valued variables with durative actions. Learning operators in this domain requires combining the following capabilities simultaneously:

– Learning symbolic literals, whose arguments can take values from a predefined collection, such as vehicle(veh26).
– Learning continuous-valued effects, such as y = f(30), fuel(veh26, y), indicating that veh26 fuel level changes as a function of f.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 86–96, 2021.
https://doi.org/10.1007/978-3-030-55180-3_7

– Learning under noise data, e.g. a state with the literal position(veh26, loc2) when veh26 is at location loc3. Noise can be introduced when a human operator creates the traces.
– Use of simulation to learn and test the resulting actions. While the learner is independent of the domain, we test the learned operators with the simulation from which we learn them.

In this paper we present an automated action-learning system that tackles these learning requirements simultaneously. To the best of our knowledge, this is the first work on learning action models combining symbolic literals and continuous effects under noisy training data.

The remainder of the paper is as follows: we first discuss related work; next we present the waypoint-navigation simulation used to generate the training data and to test the results of the learner; afterwards we discuss a description of the symbolic-learning procedure; next, we discuss a description of the procedure to learn numeric-durative preconditions and effects, accompanied by an example learned operator; then, we discuss a description of the experiments used to test the learned operators and the results; and finally, discuss future directions.

2 Related Work

Arora *et al.* [4] summarizes the current available algorithms for learning actions. The two algorithms most similar to our own in terms of learning numeric elements of action models are PlanMiner-O2 [5] and ERA [6], however both algorithms differ with respect to our work in substantial ways. ERA learns a mapping of potentially noisy inputs to a set of numeric categories corresponding to a discrete set of output values (one value per category). They use 4 categories for the input: 1-road, 2-grass, 3-dirt and 4-rocks. The sensors might return a reading such as 1.5, indicating that the terrain is either road or grass. The classification scheme might learn that an input of 1.5 should map to the return value for category 1, meaning a road; and based on the classified category, the algorithm returns the speed that a vehicle will be able to travel in that kind of terrain. Succinctly, ERA will learn the effects as a constant function $f(x_1, \ldots, x_n) = c$, where $x_1, \ldots, x_n$ are input variables and c is computed as the mean of all values of training examples with input $x_1, \ldots, x_n$. PlanMiner-O2 learns numeric preconditions, and in the effects increases or decreases numeric values by a constant factor. Succinctly, it will learn a function $f(x_1) = x_1 \pm c$ for an input variable x_1. In contrast to these two works, we are learning the numeric preconditions and effects as a continuous-valued function $f(x_1, \ldots, x_n)$ on input numeric variables $x_1, \ldots, x_n$ in addition to the symbolic preconditions.

In addition to the discussions of PlanMiner-O2 and ERA, Arora *et al.* [4] presents an overview of some 30 other works on learning action models. Our overall claim of novelty versus existing work is that this is the first action-learning system that combines learning symbolic literals and durative effects under noisy training data.

Our approach for learning the symbolic preconditions builds on ideas from the ARMS operator-learning algorithm [7], which also uses MAX-SAT. Unlike ARMS, we assume full state observability as given by the simulation. As a result, we did not need to formulate

hypothesizing causal links between actions. Thus, instead of action traces $s_0\ a_1\ s_1\ \ldots\ s_n$ partially annotated with intermediate states, s_i, the input to MAX-SAT in our algorithm are collections of triples (s, a, s'), where s and s' are the (complete) observed states before and after executing a. As a result, with an adequate threshold Θ and sufficient training examples, the false positives drop to nearly zero and the false negatives to zero. The crucial difference is that our algorithm also learns durative actions whereas ARMS learns symbolic literals only.

The EXPO system was designed to learn missing preconditions and effects or even complete operators [8]. EXPO does this by having expectations of when an action should (or should not) be solvable (respectively when a plan should or should not be generated). When a discrepancy occurs between the agent's expectations of the actions and the observed effects of the action, it examines the history of the action's applications to fix it. For instance, looking for missing conditions when the action was successfully applied by accident. Such a process is complementary to our work, when the operators could be further refined after learning the initial collection of operators. Unlike our work, EXPO is strictly symbolic and was not designed to deal with noise in the training data.

The TRAIL system [9] follows similar ideas of completing operators but it does so in a non-deterministic domain where there can be multiple outcomes. It does so by maintaining possible execution trees and assuming a teacher is available that can provide examples on-demand, allowing the learner to extrapolate missing knowledge. It can model durative actions by projecting the effects over multiple time steps and learn the projected intervals for numeric values. It requires substantial background knowledge including the goals that the agent is pursuing, used for performing inductive logic programming [10], and the means to generate traces even when the actions are incomplete by using teleo-reactive trees [11]. The most important differences versus our work is twofold. First, we can deal with noisy training data. Second, we are able to learn the changes in values of durative actions as functions.

Pasula *et al.* [12] presented an action learning system for stochastic domains and tested the learned actions for MDP planning tasks. It can deal with noise because of the stochastic nature of the domain. In our case, we can cope with noise even though the learned action model is deterministic. Furthermore, we learn durative effects.

Lindsay and Gregory [13] did a study on the domains used in the international planning competition (IPC) and identified a collection of numeric constraints that are typically used in those competitions. This collection is used to learn numeric conditions by using negative and positive examples. The negative examples are used to eliminate constraints that are inconsistent with the training data and the positive examples are used to tune the resulting constraints. In our case, we are learning the continuous function $f(x_1, \ldots, x_n)$ on input numeric variables $x_1, \ldots, x_n$ in addition to the symbolic preconditions.

Walsh and Littman's [14] system learns STRIPS operators augmented with the Web description language so they can be used for semantic web service composition. In contrast to our work, it assumes no noise in the training data and no durative effects. It also provides bounds to the operator learning problem: in the general case, learning preconditions may require an exponential number of plan prediction mistakes (PPMs). Informally, a PPM counts the number of times the operator was applied incorrectly. If the maximum number of preconditions of operators is known to be bounded by k, then the number PPM is bounded by a polynomial on k.

3 Waypoint Navigation Simulation

We implemented a simulator for agents to perform logistics and transportation tasks using the python discrete event simulation framework *SimPy*.[1] The simulator models agents or vehicles that can travel to various locations, via a network of roads (edges) that connect them. The agents navigate between waypoints in a geographic area represented by a two-dimensional coordinate system or grid as shown in Fig. 1. It depicts 17 locations on a road network or graph. Each location is marked as a vertex on the network and labelled as *loc1* through *loc17*. The edges between the vertices indicate valid roads that can be travelled. The graph is not complete and travel from one location to another may require multiple transits. Depending on agent speed, traveling between waypoints can take different times.

The simulator supports multiple vehicles to concurrently perform logistics actions such as trucks performing activities in tandem. The vehicles can perform two actions: transit and refuel. A transit action enables a vehicle to move between locations if an edge connects them. A refuel action refills a vehicle's fuel tank if it is in a location. We model these activities with continuous valued variables. Vehicles have average travel speed and fuel consumption rate. The simulator consumes inputs in the form of world states comprising vehicles, locations, and roads along with a schedule comprising list of transit and refuel actions. The simulator generates connection networks randomly.

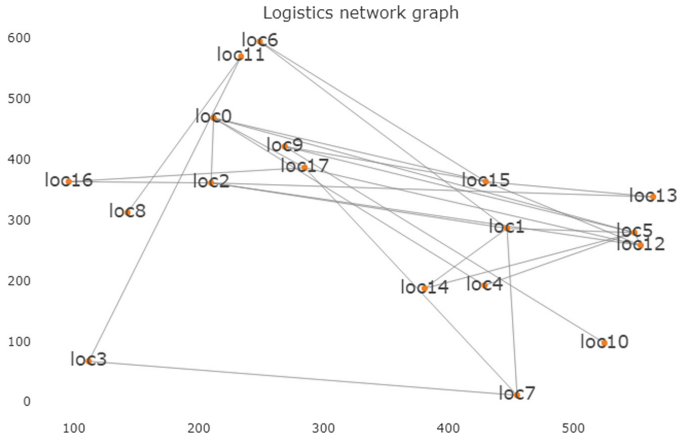


Fig. 1. Example logistics road network.

The simulator executes the input schedule and outputs a simulation trace in commonly-used JavaScript Object Notation (JSON) format[2] as shown in Fig. 2. It shows a fragment of the simulation trace obtained by executing a transit action from *loc12* to *loc5*. Every time an action starts or finishes the entire simulation state is logged into the trace as training data and a means of replaying the simulation. The trace shows the

[1] https://simpy.readthedocs.io/en/latest/.
[2] https://www.json.org/.

completion of the transit action for *veh0*; *veh0* is now at position 5. Similarly, *veh1* is at an edge between *loc6* and *loc1*. The trace notes the status of each vehicle such as rate of fuel consumption, current gas level, and total gas capacity. It enforces domain constraints at runtime and will return a failed trace if an invalid action is attempted.

4 Overview of Learning Algorithm

There are two requirements for the learning algorithm: (1) positive and negative examples and (2) background knowledge on numeric values used to infer numeric-durative effects. From these requirements as a starting point, the procedure performs the following steps: (1) simulate positive examples by randomly generating a road network and performing random walks from locations in the network; (2) Convert JSON traces into predicate form; (3) Convert numeric arguments into constants to prepare for learning symbolic literals; (4) Construct clauses (which enforcing constraints) from positive examples and run these clauses on a MAXSAT solver to learn the operator's symbolic literals; (5) Generate negative examples; (6) Use the positive examples, negative examples and background knowledge to generate training data for a logistic regression learner to generate the operator's durative effects.

We provided an example of the JSON output in Fig. 2. The conversion into predicate form is mostly straightforward and we omit the details for the sake of space.


```
"action": {
    "action_id": "17",
    "end_location": "loc5",
    "start_location": "loc12",
    "type": "TransitAction",
    "vehicle": "veh0"},
  "state": {
    "edges": [{
      "end": {"name": "loc5"},
      "name": "edgeloc0loc5",
      "start": {"name": "loc0"},
      "traffic_speed": 60.0 }, ...],
    "locations": [{"name": "loc5", "x": 549, "y": 279}, ...]
    "vehicles": [{
      "gas_tank_capacity": 30,
      "gas_tank_level": 17,
      "gph": 1.0,
      "name": "veh1",
      "position": "edgeloc6loc1"}, {
      "gas_tank_capacity": 30,
      "gas_tank_level": 19.56,
      "gph": 1.0,
      "name": "veh0",
      "position": "loc5"}]},
"status": "finished", "time": 26.0
```


Fig. 2. Example JSON simulation trace fragment.

5 Learning Symbolic Literals

5.1 MAX-SAT

The conjunctive normal form CNF-SAT is the decision problem of determining the satisfiability of a given CNF $\phi = (D_1 \curlywedge D_2 \curlywedge \ldots \curlywedge D_n)$, where each clause D_i is a disjunction $(X_1 \curlyvee X_2 \curlyvee \ldots \curlyvee X_m)$ and each literal X_j is either a Boolean variable or its negation. CNF-SAT is NP-complete. MAX-SAT is a variation of CNF-SAT in that each clause D_i has associated a nonnegative weight, $w(D_i)$. The MAX-SAT problem is defined as follows: given a CNF ϕ, find a *subset* of clauses, $D_{1'}, D_{2'}, \ldots, D_{m'}$, in ϕ such that (1) there is an assignment of the Boolean variables making each of these clauses true and (2) $W' = w(D_{1'}) + w(D_{2'}) + \ldots + w(D_{m'})$ is maximized. That is, for any other subset of clauses $D_{1''}, D_{2''}, \ldots, D_{m''}$, satisfying Condition 1, then $W' \geq W''$, where $W'' = w(D_{1''}) + w(D_{2''}) + \ldots + w(D_{m''})$. MAX-SAT is NP-hard. MAX-SAT solvers run in polynomial-time on the number of clauses and approximate a solution to find a collection of clauses that can be satisfied together with the variable Boolean assignments that make this possible. They provide no guarantee that Condition (2) is satisfied. In our work we use the MAX-SAT solver reported in Ansótegui *et al.* [15].[3]

5.2 Using MAX-SAT to Learn Symbolic Literals

We learn symbolic preconditions from positive examples, which we extract from simulator-generated random-walk traces. Positive examples are state-action-state triples, $s_t\ a_t\ s_{t+n}$, where s_t occurs immediately before the action starts, and s_{t+n} after the action ends. $n \geq 1$ because actions are durative and can interleave, so they may take several states before completion. We replace numeric values with the generic constant *num*, during symbolic learning. We compile potential preconditions from the literals in s_t of each example. We remove unlikely preconditions, namely, literals (1) which share no arguments with the action's signature or (2) which do not appear in more than some fraction, Θ of examples. Condition (1) is a relaxation of the standard STRIPS notation where arguments of literals appearing in the preconditions must be arguments named in the action's signature. Condition (1) requires at least one of the arguments of each precondition of the action TransitAction(v21, l4, l7) to be either v21, l4, or l7.

We use the state-variable action representation. That is, a state is defined as a collection of variables and their assigned values. From state to state, the variable remains the same but their values might change. For instance, the literal *position(v1, loc1)*, represents the state variable "position of vehicle v1" and indicates it is currently assigned to *loc1*.

Since actions are durative, state changes for multiple actions can occur between s_t and s_{t+n} and removing literals helps ensure that no erroneous preconditions are learned. For example,

s_t: position(v1, loc1), position(v2, loc1), ….
a: action(v1, loc1, loc2)
s_{t+n}: position(v1, loc2), position(v2, edge13), …

[3] https://github.com/jponf/wpm1py.

Although *position(v2, loc1)* could be identified as a precondition because it includes *loc1*, unless it occurs in s_t of multiple examples, it will be removed from consideration.

For each remaining literal, we include a clause in our CNF equation asserting that the literal is a precondition; this means that the literal is a candidate to be a precondition of the action and it will be up to MAX-SAT to determine which of those clauses are true. The weight of the clause is the number of examples for which the literal appears in s_t. We add additional clauses in our CNF encoding to enforce the following conditions:

1. If a state-variable in s_{t+n} has a changed value compared to s_t, it must be a true precondition.
2. A state-variable in the effects must have changed its value with respect to its preconditions.
3. All actions have at least one precondition and one effect.

Constraint 1 ensures that the preconditions include literals relevant to the action parameters when the state-variable changed from s_t to s_{t+n}. Constraint 2 requires that all effects are literals that underwent changes during the action. Constraint 3 ensures that all actions are non-trivial, i.e., that they effect some change in the state.

6 Learning Durative Preconditions and Effects

Unlike symbolic literals, learning durative effects requires positive and negative examples and background information. We learn a logistic regression model [16] for *gas_tank_level* as a function of the initial tank level and the travel distance, which we use in both preconditions and effects.

Domain Background Information. To compute the gas tank level of a vehicle we need the distance between locations. However, distance is not an explicit attribute in the state, so we provide background information indicating how to calculate distance along an edge by computing the Manhattan distance between its start and end locations.

6.1 Using Logistic Regression

For learning numeric literals, we revert back from the generic constant *num* to the original numeric values in all positive examples. To generate negative examples, we generate traces using random walks in the simulator and remove all refuel actions from the traces. We simulate these modified traces and if for a particular state-action pair, s_i a_{i+1}, in the trace, the action is not executable by the simulator, then the pair (s_i, a_{i+1}) is used as a negative example for action a_{i+1}.

From each example (negative or positive), we extract numeric preconditions and discard any where the value is the same across all examples, e.g., the gallons per hour, *gph*. We use the remaining values (x and y for start location and end location, *gas_tank_level* for vehicle) and the background information to calculate the distance between the locations and train a Logistic Regression model on distance and *gas_tank_level*, using the positive and negative labels of the examples as our classes. In the effects, we calculate the new value for *gas_tank_level* using the decision function of the trained logistic regression model.

6.2 Example Learned Operator

Figure 3 shows the learned transit operator, TransitAction. The vehicle *?vehicle* will move from location *?start* to location *?end* along edge *?edge* (question marks denoted variables). The start and end of *?edge* are defined in two literals. The effects describe the change in the vehicle's position and the gas tank level.

Head:
TransitAction(?vehicle, ?start, ?end)
Preconditions:
gas_tank_capacity(?vehicle, ?gtcv)
gas_tank_level(?vehicle, ?gtlv1)
gph(?vehicle, ?gphv)
position(?vehicle, ?start)
type(?vehicle, "vehicles")
end(?edge, ?end)
type(?end, "locations")
x(?end, ?xe)
y(?end, ?ye)
start(?edge, ?start)
type(?start, "locations")
x(?start, ?xs)
y(?start, ?ys)
?gtlv2 ← -0.0328 + 0.62*?gtlv1 + -0.199*distance_metric((?xs, ?ys), (?xe, ?ye))
?gtlv2 > 0
Effects:
position(?vehicle, ?end)
gas_tank_level(?vehicle, ?gtlv2)

Fig. 3. Learned TransitAction operator.

7 Empirical Evaluation

We perform two sets of analysis, the first evaluating the effectiveness of MAX-SAT at learning the symbolic preconditions, the second testing the effectiveness of the whole learning procedure.

7.1 Experimental Setup

To evaluate the effectiveness of the symbolic literal learning component, we generate a pool of 100 simulations, each with a different initial state, containing at least one of each type of action. From each simulation we extract one positive example of each action type to create a pool of 100 positive examples for each action. From this pool of 100, we select 5 sets of 3, 10, and 30 examples and learn the symbolic preconditions for each set. For each set, we also select a set of mislabeled examples to test the effects of incorrect examples (noise) on the system's learning ability. Mislabeled examples are

examples which do not contain the appropriate symbolic preconditions, but are treated as positive learning examples. We learn preconditions from each of these sets at three different Θ values, 0.1, 0.2, and 0.3. The performance metrics were false positives (i.e., additional preconditions, not appearing in the ground truth) and false negatives (i.e., missing preconditions). The ground truth is a collection of literals that were manually selected in order to ensure that the operator was correctly learned.

To test the effectiveness of the whole procedure, we select a set of 10 positive examples, 8 negative examples, and the corresponding learned preconditions. We create a set of operators for each Θ value and test these operators on 30 randomly-generated planning problems. The problems are constructed by generating a random state and randomly selecting a vehicle and destination location. The performance metric is the number of test problems for which a valid plan is generated; plan validity is tested by running the plan on the simulator.

Solution plans are generated using HTN planning techniques, which direct the selected vehicle to explore the state map until it arrives at the goal location, refueling to ensure a full tank before each transit. To validate a generated plan, we convert to JSON schedule format and simulate from the start state.

7.2 Results

Figure 4 shows the false positive rate for learning symbolic preconditions is non-zero and dependent on both the Θ value and the number of learning examples. Each line represents the combined false positive rate (FPR) for symbolic preconditions for both operators when learning from a set of positive examples and/or a set of positive examples with an additional 10% of examples which are mislabeled, simulating noise in the data.

The FPR changes more between 3 and 10 positive examples than it does between 10 and 30 examples. When learning from 10 or more examples, the largest improvement occurs between theta of 0.1 and 0.2, with little improvement between 0.2 and 0.3.

Interestingly, noise added via mislabeled examples had little impact on the number of false positive preconditions, except at $\Theta = 0.3$ for 3 positive examples. The addition of mislabeled examples actually improves the set of preconditions, because it decreases the percent of examples in which the same non-preconditions occur, allowing more non-preconditions to be filtered out.

Even for $\Theta = 0.3$ and 30 training examples, there is a non-zero error rate in the false positives because literals (e.g., *gph*) were the same throughout the examples and the learner acquired them, however we deemed them superfluous because they were constant and present in all training and testing examples.

There were no false negative literals learned in any of the runs, demonstrating the effectiveness of the MAX-SAT procedure.

Figure 5 shows the number of test problems (out of 30) for which the planner is able to generate plans using operators learned with $\Theta = 0.1$, 0.2 and 0.3. The simulator validated all generated plans. For $\Theta = 0.1$, slightly more than half of the test problems were solvable by the planner. As we can see from Fig. 4, with $\Theta = 0.1$ and 30 examples, there are some 15% additional literals in the preconditions, restricting applicability of operators.

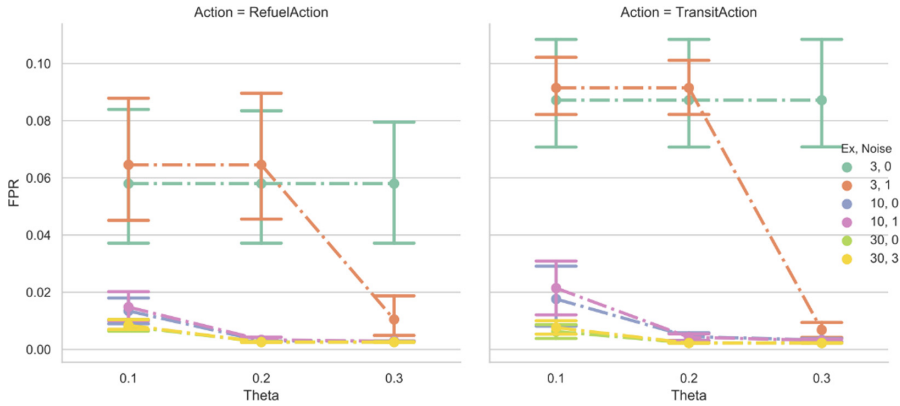


Fig. 4. False positive rates with respect for the Refuel and the Transit action.

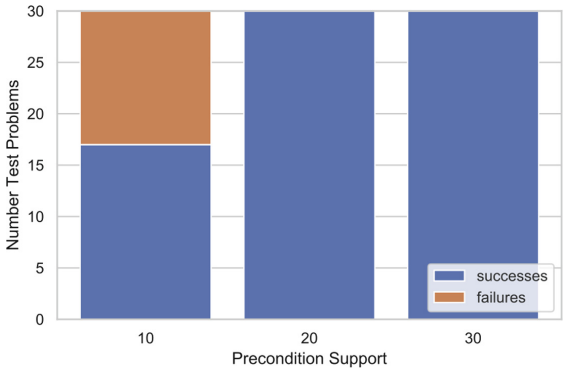


Fig. 5. Number of solved test problems for $\Theta = 10\%$, 20% and 30%. Solution plans were run in the simulator. All test problems were solved and verified for $\Theta = 20\%$ and 30% and more than half were for $\Theta = 10\%$.

8 Final Remarks

Our algorithm successfully learns operators with symbolic literals and continuous effects even under noisy training data. To the best of our knowledge, this is the first action learning algorithm that combines symbolic and numeric fluents and durative effects.

There are a number of possible future directions. First, we provide background knowledge because our simulator doesn't compute the distance explicitly as part of the state information. It would have been easy to modify the simulator and make the distance explicit. Indeed, other waypoint navigation simulators explicitly compute distances. However, in general some kind of background knowledge is needed when computing complex numeric literals, such as durative effects, on conditions such as traffic flow, traffic speed etc. In the future, we want to explore inductive learners such as ILASP which may help induce background knowledge. Second, we want to take into account temporal considerations such as an action starting at time t and a second action starting

at a time $t + \Delta$ and the effects of both actions contributing towards a third action starting later. For instance, at time t, a hose starts adding water to a container at a certain fixed rate, and later at time $t + \Delta$ another hose starts adding water to the same container. Later we open a faucet to drain the container. In this case the rate that the container is filled and drained is a function of time.

References

1. Bettstetter, C., Wagner, C.: The spatial node distribution of the random waypoint mobility model. In: WMAN, vol. 11, pp. 41–58 (2002)
2. Tan, C.H., Ang, J.H., Tan, K.C., Tay, A.: Online adaptive controller for simulated car racing. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2239–2245. IEEE (2008)
3. Bruce, J., Veloso, M.M.: Real-time randomized path planning for robot navigation. In: Robot Soccer World Cup, pp. 288–295. Springer, Heidelberg (2002)
4. Arora, A., Fiorino, H., Pellier, D., Métivier, M., Pesty, S.: A review of learning planning action models. Knowl. Eng. Rev. **33** (2018)
5. Segura-Muros, J.Á., Pérez, R., Fernández-Olivares, J.: Learning numerical action models from noisy and partially observable states by means of inductive rule learning techniques. In: KEPS 2018, vol. 46 (2018)
6. Balac, N., Gaines, D.M., Fisher, D.: Learning action models for navigation in noisy environments. In: ICML Workshop on Machine Learning of Spatial Knowledge, Stanford, July 2000
7. Yang, Q., Wu, K., Jiang, Y.: Learning action models from plan examples using weighted MAX-SAT. Artif. Intell. **171**(2–3), 107–143 (2007)
8. Gil, Y.: Learning by experimentation: incremental refinement of incomplete planning domains. In: Machine Learning Proceedings 1994, pp. 87–95. Morgan Kaufmann (1994)
9. Benson, S.S.: Learning action models for reactive autonomous agents. Doctoral dissertation, Stanford University (1996)
10. Lavrac, N., Dzeroski, S.: Inductive logic programming. In: WLP, pp. 146–160 (1994)
11. Nilsson, N.: Teleo-reactive programs for agent control. J. Artif. Intell. Res. **1**, 139–158 (1993)
12. Pasula, H.M., Zettlemoyer, L.S., Kaelbling, L.P.: Learning symbolic models of stochastic domains. J. Artif. Intell. Res. **29**, 309–352 (2007)
13. Lindsay, A., Gregory, P.: Discovering Numeric Constraints for Planning Domain Models. In: KEPS 2018, vol. 62 (2018)
14. Walsh, T.J., Littman, M.L.: Efficient learning of action schemas and web-service descriptions. In: AAAI-2008, vol. 8, pp. 714–719 (2008)
15. Ansótegui, C., et al.: Improving SAT-based weighted MaxSAT solvers. In: International Conference on Principles and Practice of Constraint Programming, pp. 86–101. Springer, Heidelberg (2012)
16. Hosmer Jr., D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression, vol. 398. Wiley, Hoboken (2013)

Understanding and Exploiting Dependent Variables with Deep Metric Learning

Niall O'Mahony(✉), Sean Campbell, Anderson Carvalho, Lenka Krpalkova, Gustavo Velasco-Hernandez, Daniel Riordan, and Joseph Walsh

IMaR Research Centre, Institute of Technology Tralee, Tralee, Ireland
niall.omahony@research.ittralee.ie

Abstract. Deep Metric Learning (DML) approaches learn to represent inputs to a lower-dimensional latent space such that the distance between representations in this space corresponds with a predefined notion of similarity. This paper investigates how the mapping element of DML may be exploited in situations where the salient features in arbitrary classification problems vary over time or due to changing underlying variables. Examples of such variable features include seasonal and time-of-day variations in outdoor scenes in place recognition tasks for autonomous navigation and age/gender variations in human/animal subjects in classification tasks for medical/ethological studies. Through the use of visualisation tools for observing the distribution of DML representations per each query variable for which prior information is available, the influence of each variable on the classification task may be better understood. Based on these relationships, prior information on these salient background variables may be exploited at the inference stage of the DML approach by using a clustering algorithm to improve classification performance. This research proposes such a methodology establishing the saliency of query background variables and formulating clustering algorithms for better separating latent-space representations at run-time. The paper also discusses online management strategies to preserve the quality and diversity of data and the representation of each class in the gallery of embeddings in the DML approach. We also discuss latent works towards understanding the relevance of underlying/multiple variables with DML.

Keywords: Deep Metric Learning · Variable features · Dependent variables · Computer vision

1 Introduction

Deep Learning has great achievements in computer vision for various classification and regression tasks in terms of accuracy, generalisability and robustness. However, to achieve this performance require training on hundreds or thousands of images and very large datasets. Fine-tuning these models for fine-grained visual recognition tasks is not always straightforward however and has prompted the creation of a type of architecture for this type of problem known as metric learning. Metric Learning is popular in Computer Vision for tasks such as face verification/recognition [1], person re-identification

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 97–113, 2021.
https://doi.org/10.1007/978-3-030-55180-3_8

[2, 3], 3D shape retrieval [4] and landmark recognition [5] and is also used in other fields, e.g. for Question Paraphrase Retrieval in Speech Recognition [6], music classification [7] and bioacoustic classification [8] from audio data and gesture recognition from accelerometer data [9]. In Sect. 2, we will further define the research problems relevant to our research and in Sect. 3 we will introduce the background theory of Metric Learning for the reader.

The challenges of fine-grained visual recognition relate to two aspects: inter-class similarity and intra-class variance. In Sect. 4, this paper will review some methodologies which have been proposed in recent research to optimize these two attributes of the embedding space of DML, e.g. through learning dependent relationships in the fields of multi-label classification and newly proposed cost functions, and also methods which exploit the embedding space for interpreting the inner workings of the neural network. In Sect. 5, this paper will also propose a unique approach to improving classification accuracy of DML in any arbitrary applications through the injection of apriori knowledge of dependent variables into a clustering algorithm appended to the inference pipeline of the DML approach. Examples of such variable features include seasonal and time-of-day variations in outdoor scenes in place recognition tasks for autonomous navigation [5], age/gender variations in human/animal subjects in medical/ethological studies [10] and operator/time-of-shift variations in industrial automation tasks. We will also propose an online management strategy to preserve the quality and diversity of data and the representation of each class in the gallery of embeddings in the DML approach. Finally, in Sect. 6, this paper will conclude with a discussion of our findings to date and of future work which is currently being actively engaged in follow-up this work.

2 Problem Definition

In the field of deep learning, the quality of input data is often more important than the model architecture and training regimen. The challenges of dataset management include ensuring the dataset is correctly labelled, balanced and contains a sufficient amount of data. As well as this, the categories to be classified must also be chosen carefully at the task definition stage to minimize intra-class variance, i.e. it is harder to train a deep learning network to reliably classify 'animals' than it is to train one to classify just 'cats' or 'dogs'. However, breaking down the categories to a low enough level can be difficult, requiring the judgement of an application expert and may introduce unwanted bias. Furthermore, system maintenance does not end once the problem is defined and the model is trained. In situations where salient features to the classification problem vary depending on auxiliary variables, it would be useful to leverage these auxiliary variables (if they are known apriori to classification) to narrow down the classification results to instances which are more likely in light of this new knowledge.

2.1 One-Shot Learning

The term One-Shot Learning represents a still-open challenge in computer vision to learn much information about an object category from just one image. Few-shot and zero-shot learning are similar classification problems but with different requirements on

how many training examples are available. Few-shot learning, sometimes called low-shot learning often falls under the category of OSL and denotes that multiple images of new object categories are available rather than just one. Zero-shot learning algorithms aim at recognizing object instances belonging to novel categories without any training examples [11]. The motivation for this task lies not only in the fact that humans, even children, can usually generalize after just one example of a given object but also because models excelling at this task would have many useful applications. Example applications include facial recognition in smart devices, person re-identification in security applications as well as miscellaneous applications across industry, e.g. fine-grained grocery product recognition by [13], drug discovery in the pharmaceutical industry [12], stable laser vision seam-tracking systems [13] and the detection of railway track switches, face recognition for monitoring operator shift in railways and anomaly detection for railway track monitoring [14].

If it is desired for a conventional machine learning classifier to identify new classes on top of those it was trained to classify then the data for these classes must be added to the dataset (without unbalancing the dataset) and the model must be retrained entirely. This is why metric learning is so useful in these situations where information must be learnt about new object categories from one, or only a few, training samples. The general belief is that gradient-based optimization in high capacity classifiers requires many iterative steps over many examples to perform well. This type of optimization performs poorly in the few-shot learning task.

In this setting, rather than there being one large dataset, there is a set of datasets, each with few annotated examples per class. Firstly, they would help alleviate data collection as thousands of labelled examples are not required to attain reasonable performance. Furthermore, in many fields, data exhibits the characteristic of having many different classes but few examples per class. Models that can generalize from a few examples would be able to capture this type of data effectively.

Gradient descent-based methods weren't designed specifically to perform well under the constraint of a set number of updates nor guarantee speed of convergence, beyond that they will eventually converge to a good solution after what could be many millions of iterations. Secondly, for each separate dataset considered, the network would have to start from a random initialization of its parameters.

Transfer learning can be applied to alleviate this problem by fine-tuning a pre-trained network from another task which has more labelled data; however, it has been observed that the benefit of a pre-trained network greatly decreases as the task the network was trained on diverges from the target task. What is needed is a systematic way to learn a beneficial common initialization that would serve as a good point to start training for the set of datasets being considered. This would provide the same benefits as transfer learning, but with the guarantee that the initialization is an optimal starting point for fine-tuning [15].

Over years many algorithms have been developed in order to tackle the problem of One-shot learning including:

- Probabilistic models based on Bayesian learning [16, 17],
- Generative models using probability density functions [18, 19],
- Applying transformation to images [20, 21],

- Using memory augmented neural networks [22],
- Meta-learning [15, 23] and
- Metric learning.

This paper will focus on the metric learning approach because of the way it learns to map it's output to a latent space and how this may be exploited to infer relationships between feature variability and auxiliary background information.

2.2 Fine-Grained Visual Categorization

Fine-grained visual categorization (FGVC) aims to classify images of subordinate object categories that belong to a same entry-level category, e.g., different species of vegetation [24], different breeds of animals [25] or different makes of man-made objects [26].

The visual distinction between different subordinate categories is often subtle and regional, and such nuance is further obscured by variations caused by arbitrary poses, viewpoint change, and/or occlusion. Annotating such samples also requires professional expertise, making dataset creation in real-world applications of FGVC expensive and time-consuming. FGVC thus bears problem characteristics of few-shot learning.

Most existing FGVC methods spend efforts on mining global and/or regional discriminative information from training data themselves. For example, state-of-the-art methods learn to identify discriminative parts from images of fine-grained categories through the use of methods for interpreting the layers of Convolutional Neural Networks, e.g. Grad-CAM [27]. However, the power of these methods is limited when only few training samples are available for each category. To break this limit, possible solutions include identifying auxiliary data that are more useful for (e.g., more related to the FGVC task of interest, and also better leveraging these auxiliary data. These solutions fall in the realm of domain adaptation or transfer learning and the latter has been implemented by training a model to encode (generic) semantic knowledge from the auxiliary data, e.g. unrelated categories of ImageNet, and the combined strategy of pretraining followed by fine-tuning alleviates the issue of overfitting. However, the objective of pre-training does not take the target FGVC task of interest into account, and consequently, such obtained models are suboptimal for transfer. An important issue to achieve good transfer learning is that data in source and target tasks should share similar feature distributions. If this is not the case, transfer learning methods usually learn feature mappings to alleviate this issue.

Alternative approaches include some of those listed for one-shot learning above. Meta-learning has been adopted by [28] to directly identify source data/tasks that are more related to the target one, i.e. select more useful samples from the auxiliary data and remove noisy, semantically irrelevant images. Metric learning has been used similarly during training dataset creation through partitioning training images within each category into a few groups to form the triplet samples across different categories as well as different groups, which is called Group Sensitive TRiplet Sampling (GS-TRS). Accordingly, the triplet loss function is strengthened by incorporating intra-class variance with GS-TRS, which may contribute to the optimization objective of triplet network [26].

Metric Learning has also been employed to overcome high correlation between subordinate classes by learning to represent objects so that data points from the same class

will be pulled together while those from different classes should be pushed apart from each other. Secondly, the method overcomes large intra-class variation (e.g., due to variations in object pose) by allowing the flexibility that only a portion of the neighbours (not all data points) from the same class need to be pulled together. The method avoids difficulty in dealing with high dimensional feature vectors (which require $O(d^2)$ for storage and $O(d^3)$ for optimization) by proposing a multi-stage metric learning framework that divides the large-scale high dimensional learning problem to a series of simple subproblems, (achieving $O(d)$ computational complexity) [26].

3 Metric Learning

Generally speaking, Metric learning can be summarised by the learning of a similarity function which is trained to output a representation of its input, often called an embedding. During training, an architecture consisting of several identical entities of the network being trained is used along with a loss function to minimize the distance between embeddings of the same class (intra-class variability) and maximize the space between classes (inter-class similarity) so that an accurate prediction can be made. The resulting embedding of each query input is compared using some distance metric against a gallery of embeddings which have been collected from previous queries. In this way, queries need not necessarily be in the training data in order to be re-identified, making the methodology applicable to problems such as facial authentication and person re-identification in security and other one-shot or few-shot learning applications.

Features extracted from classification networks show excellent performance in image classification, detection and retrieval, especially when fine-tuned for target domains. To obtain features of greater usefulness, end-to-end distance metric learning (DML) has been applied to train the feature extractor directly. DML skips the final SoftMax classification layer normally present at the end of CNN's and projects the raw feature vectors to learned feature space and then classifies input image based on how far they are from learned category instances as measured by a certain distance metric. Due to the simplicity and efficiency, the metric-based approach has been applied in industry for tasks like face recognition and person re-identification [29].

The metric-based methods can achieve state-of-the-art performance in one-shot classification tasks, but the accuracy can be easily influenced when the test data comes from a different distribution [13] The way metric learning works in practice is to have a general model which is good at learning how to represent object categories as 'embeddings', i.e. feature maps, in a feature space such that they all categories are spaced far enough away from each other that they are distinguishable. The second step is to compare each embedding that this model generates for the input image with the embeddings of all previously seen objects. If the two embeddings are close enough in the feature space (shown in Fig. 1) beyond a certain threshold, then the object is identified. The library of embeddings that are compared from may be updated continuously by adding successfully identified embeddings by some inclusion prioritization. If an object is not identified, an external system, e.g. a human expert, may need to be consulted for the correct object label to be applied.

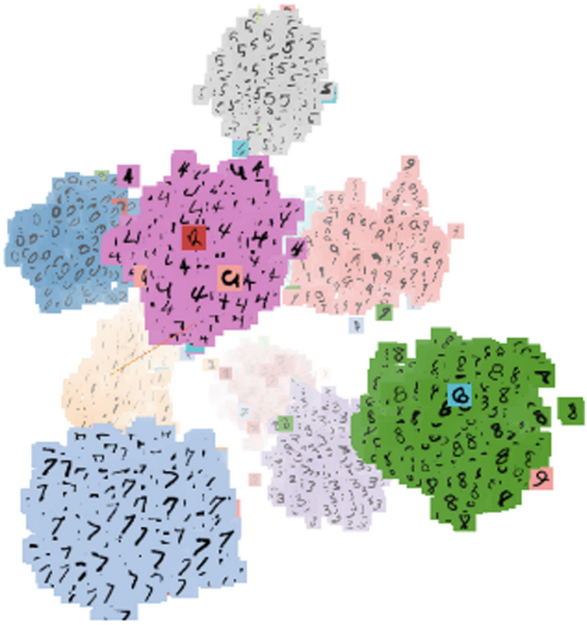


Fig. 1. A t-SNE (T-distributed stochastic neighbour embedding) visualization of a feature space used in metric learning of the MNIST dataset [30].

3.1 Distance Metrics

Two images, x_1 and x_2, are compared by computing the distance d between their embeddings $f(x_1)$ and $f(x_2)$. If it is less than a threshold (a hyperparameter), it means that the two pictures are the same object category, if not, they are two different object categories.

$$d(x_1, x_2) = f(x_1) - f(x_2) \tag{1}$$

Where f is defined as a parametric function denoting the neural network described earlier that maps high-resolution inputs (images x_1 and x_2) to low-resolution outputs (embeddings $f(x_1)$ and $f(x_2)$).

It is important to note the distance metric used as this will be used in the loss function which has to be differentiable with respect to the model's weights to ensure that negative side effects will not take place. Distance function which are often used include the Euclidean distance or the squared Euclidean distance [31], the Manhattan distance (also known as Manhattan length, rectilinear distance, L1 distance or L1 norm, city block distance, Minkowski's L1 distance, taxi-cab metric, or city block distance), dot product similarity, Mahalanobis, Minkowski, Chebychev, Cosine, Correlation, Hamming, Jaccard, Standardized Euclidean and Spearman distances [32].

3.2 Loss Functions

Loss in metric learning is defined as a measure of the distance of embeddings from sets of similar and dissimilar embeddings. For example, if two images are of the same class, the

loss is low if the distance between their associated feature vectors are low, and high if the distance between their associated feature vectors is high. Vice versa, if the two images are of different classes, the loss is only low when the image feature representations are far apart. There are many types of loss function as will become apparent in the next section which will discuss the different kinds of metric learning architecture.

3.3 Architectures

There are a number of different ways in which the base feature extractor is embedded in a metric learning architecture. By and large, the general attributes of these architectures include:

a) An ability to learn generic image features suitable for making predictions about unknown class distributions even when very few examples from these new distributions are available.
b) Amenability to training by standard optimization techniques in accordance with the loss function that determines similarity.
c) Being unreliant on domain-specific knowledge to be effective.
d) An ability to handle both sparse data and novel data.

To develop a metric learning approach for image classification, the first step is to learn to discriminate between the class-identity of image pairs, i.e. to get an estimate of the probability that they belong to the same class or different classes. This model can then be used to evaluate new images, exactly one per novel class, in a pairwise manner against the test image. The pairing with the highest score according to the network is then awarded the highest probability. If this probability is above a certain threshold then the features learned by the model are sufficient to confirm or deny the identity test image from the set of stored class identities and ought to be sufficient for similar objects, provided that the model has been exposed to a good variety of scenarios to encourage variance amongst the learned features [33].

Siamese Network

A Siamese neural network has the objective to find how similar two comparable things are and are so-called as they consist of two identical subnetworks (usually either CNNs or autoencoders), which both have the same parameters and weights as illustrated in Fig. 2. The basic approach of Siamese networks can be replicated for almost any modality.

The output of many Siamese networks are fed to a contrastive loss function, which calculates the similarity between the pairs of images (x_i and x_j). The input image x_i with samples from both similar and dissimilar sets. For every pair (x_i and x_j), if they belong to the set of similar samples S, a label of 0 is assigned to the pair, otherwise, it a label of 1 is assigned. In the learning process, the system needs to be optimized such that the distance function d is minimized for similar images and increased for dissimilar images according to the following loss function:

$$L\big(x_i, x_j, y\big) = y.d(x_1, x_2)^2 + (1 - y)\max(m - d(x_1, x_2))^2 \tag{2}$$

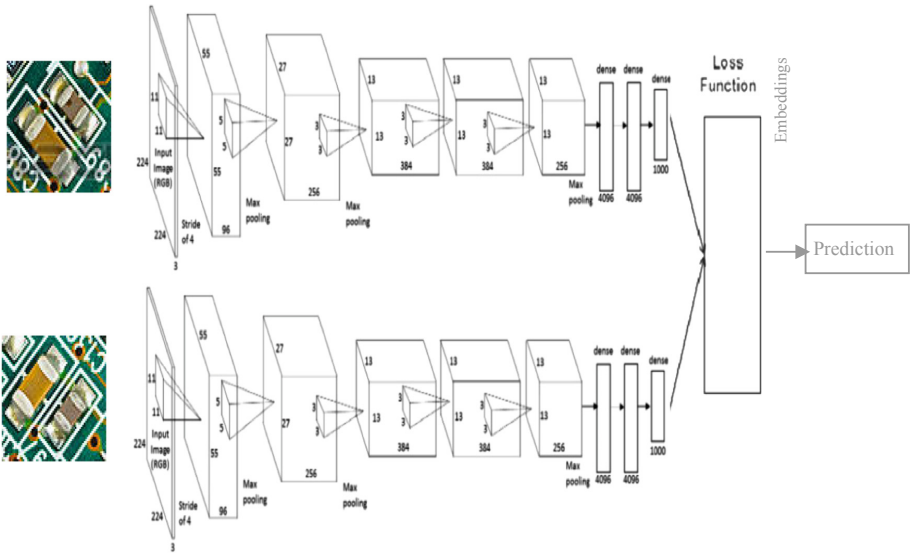


Fig. 2. Siamese network architecture

Triplet Network

The triplet loss is the key to utilize the underlying connections among instances to achieve improved performance. In a similar manner to Siamese networks, triplet networks consist of three identical base feature extractors. The triplet loss function is a more advanced loss function using triplets of images: an anchor image x_a, a positive image x_+ and a negative image x_-, where (x_+ and x_a) have the same class labels and (x_- and x_a) have different class labels. Intuitively, triplet loss encourages to find an embedding space where the distances between samples from the same classes (i.e., x_+ and x_a) are smaller than those from different classes (i.e., x_- and x_a) by at least a margin m (Fig. 3). Specifically, the triplet loss could be computed as follows:

$$Ltpl = \sum_{i=1}^{n} \max(0, m + d(x_+, x_a) - d(x_-, x_a)) \tag{3}$$

Fig. 3. The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity [34].

One advantage of the triplet loss is that it tries to be less "greedy" than the contrastive loss (which considers pairwise examples). The contrastive loss, on the other hand, only

considers pairwise examples at a time, so in a sense, it is more greedy. The triplet loss is still too greedy however since it heavily depends on the selection of the anchor, negative, and positive examples. The magnet loss introduced by [35] tries to mitigate this issue by considering the distribution of positive and negative examples. [36] compares these different loss functions and found that End-to-end DML approaches such as Magnet Loss show state-of-the-art performance in several image recognition tasks although they yet to reach the performance of simple supervised learning.

Another popular distance-based loss function is the center loss, which calculated on pointwise on 3d point cloud data. The emerging domain of geometric deep learning is an intriguing one as begin to leverage the information within 3D data. Center loss and triplet loss have been combined in the domain of 3d object detection to be able to achieve significant improvements compared with the state-of-the-art. After that, many variants of triplet loss have been proposed. For example, PDDM [37] and Histogram Loss [38] use quadruplets.

Quadruplet Network

The quadruplet network was designed on the intuition that more instances/replications of the base network as shown in Fig. 4) lead to better performance in the learning process. Therefore a new network structure was introduced by adding as many instances into a tuple as possible (including a triplet and multiple pairs) and connect them with a novel loss combining a pair-loss (which connects outputs of exemplar branch and instances branch) and a triplet based contractive-loss (which connects positive, negative and exemplar branches) [39, 40]. Beyond quadruplets, more recent works have used networks with even more instances, such as the n-pair loss [41] and Lifted Structure [38] which place constraints on all images in batches.

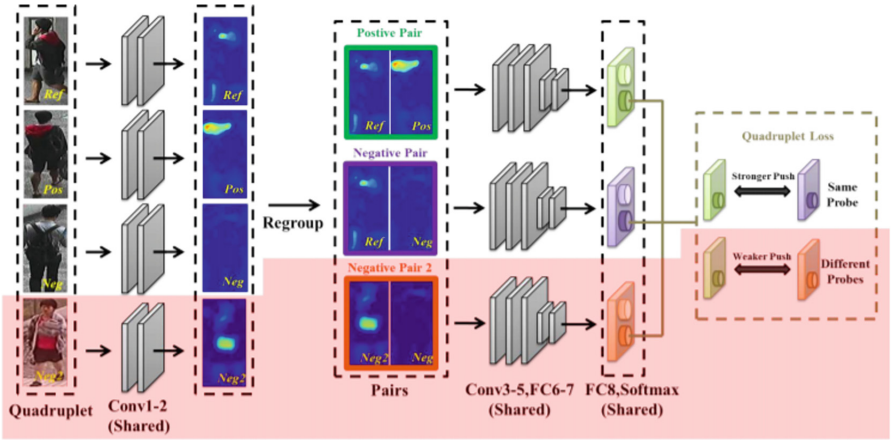


Fig. 4. Quadruplet network

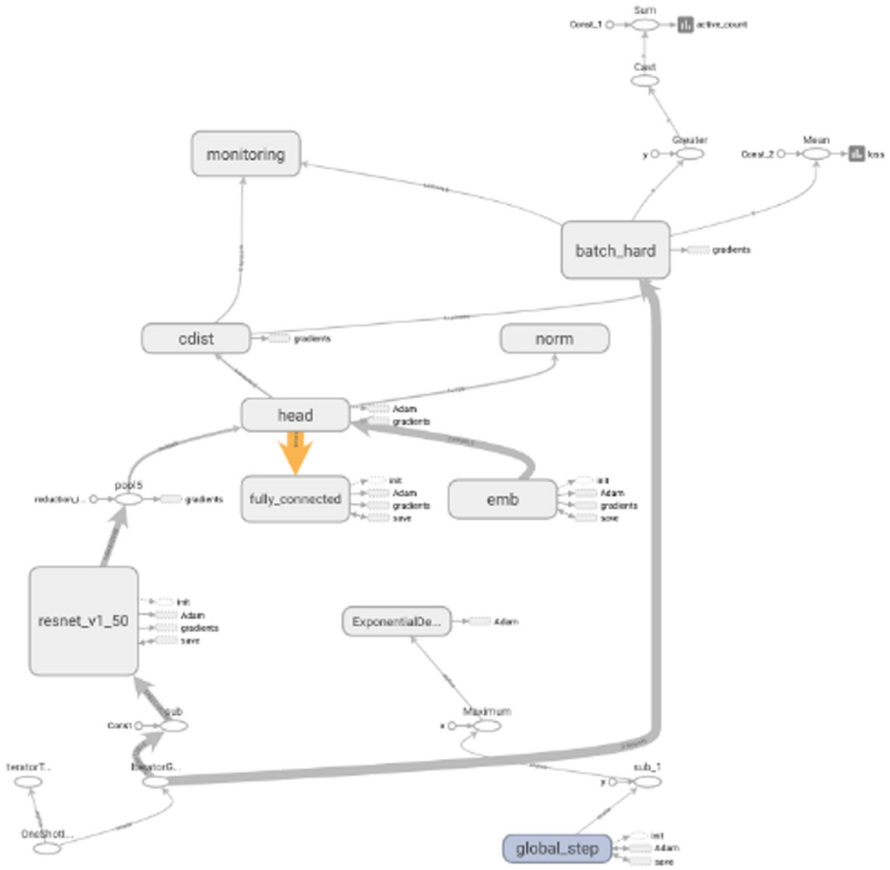


Fig. 5. The metric learning graph in tensorboard

3.4 The Head of the Network Architecture

The attributes of the network head where the replica base networks meet are also influential on performance. Networks which have been used at this stage include (e.g. which may be a fully-connected layer, a SoftMax layer or a direct throughput.

Another attribute that is controlled at the network head is the level of data augmentation. Data augmentation is a key step to ensuring the model has been exposed to sufficient variance at the training phase that is representative of the real world conditions. By rotating, blurring, or cropping image data, synthetic images can be created that approximately mirror the distribution of images in the original dataset. This method is not perfect, however —it provides a regularizing effect that may be unwanted if the network is already not performing well in training. It is worth noting that training takes significantly longer when data augmentation is applied, e.g. it takes 10 times longer if we apply flip augmentation with 5 crops of each image, because a total of 10 augmentations per image needs to be processed (2 flips times 5 crops).

Another set of hyperparameters is how the embeddings of the various augmentations should be combined. When training using the Euclidean metric in the loss, simply taking the mean is what makes the most sense. But if one, for example, trains a normalized embedding, The embeddings must be re-normalized after averaging at the aggregation stage in the head network. Figure 5 shows how the network head links these attributes.

4 Related Work

4.1 Dependent Variables

The loss function design in metric learning could be a subtle way of dealing with high degrees of variance due to dependent variables. The contrastive loss pulls all positives close, while all negatives are separated by a fixed distance. However, it could be severely restrictive to enforce such a fixed distance for all negatives. This motivated the triplet loss, which only requires negatives to be farther away than any positives on a per-example basis, i.e., a less restrictive relative distance constraint. However, all the aforementioned loss functions formulate relevance as a binary variable. The use of a ladder loss has been proposed by [42] to extend the triplet loss inequality to a more general inequality chain, which implements variable push-away margins according to respective relevance degrees measured by a proper Coherent Score metric.

4.2 Multi-label/Multi-feature/Multi-task Learning

Multi-task learning can be seen as a form of inductive transfer which can help improve a model by introducing inductive bias. The inductive bias in the case of multi-task learning is produced by the sheer existence of multiple tasks, which causes the model to prefer the hypothesis that can solve more than one task. Multi-task learning usually leads to better generalization [43]. Multi-label metric learning extends metric learning to deal with multiple variables with the same network. Instances with the more different labels are spread apart, but ones with identical labels will concentrate together. Therefore, introducing more variables means that the latent space is distributed in a more meaningful way in relation to the application domain

It has been proposed in recent work that multiple features should be used for retrieval tasks to overcome the limitation of a single feature and further improve the performance. As most conventional distance metric learning methods fail to integrate the complementary information from multiple features to construct the distance metric, a novel multi-feature distance metric learning method for non-rigid 3D shape retrieval which can make full use of the complementary geometric information from multiple shape features has been presented [4].

An alternative formulation for multi-task learning has been proposed by [44] who use a recent version of the K Nearest Neighbour (KNN) algorithms (large margin nearest neighbour) but instead of relying on separating hyperplanes, its decision function is based on the nearest neighbour rule which inherently extends to many classes and becomes a natural fit for multi-task learning [44]. This approach is advantageous as the feature space generated from Metric Learning crucially determines the performance of the KNN algorithm, i.e. the learned latent space is preserved, KNN just solves the multi-label problem within.

5 Our Approach

5.1 Using the Latent Space to Understand Dependent Variables

Often the feature vector or embedding output is a 128×1 vector or something of that order meaning that the latent space has 128 dimensions and therefore impossible for humans to visualise. There are tools, however, for dimensionality reduction of the latent space, e.g. PCA (Principal Component Analysis) and t-SNE (T-distributed stochastic neighbour embedding) are often used to visualise latent feature spaces in 2/3 dimensions as shown in Fig. 6.

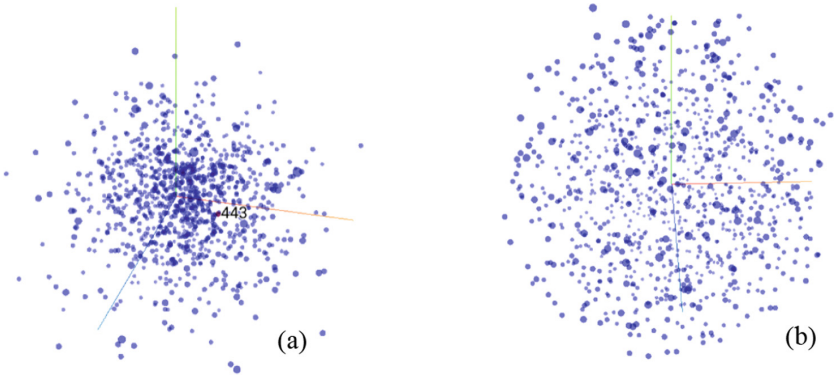


Fig. 6. (a) PCA (Principle Component Analysis) and (b) t-SNE (T-Distributed Stochastic Neighbour Embedding) projections to 3 dimensions of a latent space with 1024 embeddings. These prohjections were viewed using tensorboard [45].

Many works have used these visualisation tools to interpret the performance of the DML model, e.g. as in Fig. 7, as well as breakdown attributes of the input relevant to the application as demonstrated by [46] who map transient scene attributes a small number of intuitive dimensions to allow characteristics such as level of snow/sunlight/cloud cover to be identified in each image of a scene (Fig. 8).

5.2 Clustering in the Latent Space Based on Auxiliary Background Variables

In situations where salient features to the classification problem vary depending on auxiliary variables, it would be useful to leverage these auxiliary variables (if they are known apriori to classification) to narrow down the classification results to instances which are more likely in light of this new knowledge. Better still, if a clustering algorithm, e.g. k-means clustering, could be formulated taking as input the salient background variables and outputting a function which maps the latent space to valid classifications. For specificity, we take the example of the cross-season correspondence dataset [46]. As depicted in Fig. 9, this dataset could be used in future work to prove our proposition that clustering the latent space according to the known time of year may be used to minimise the inter-class similarity to below the acceptable threshold, τ, used at the classification stage.

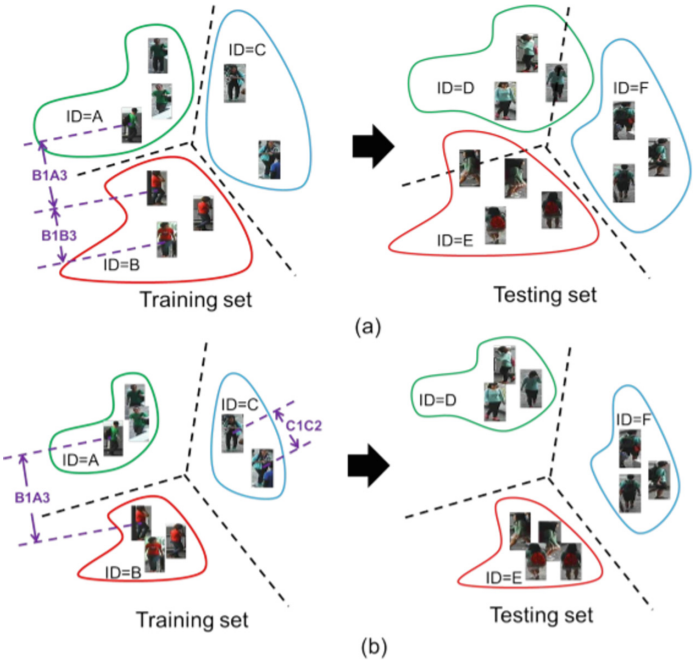


Fig. 7. A visualisation with images corresponding to each embedding as shown here in work comparing the performance of (a) triplet loss and (b) quadruplet loss and assess attributes such as l intra-class variation and a large inter-class variation [28].

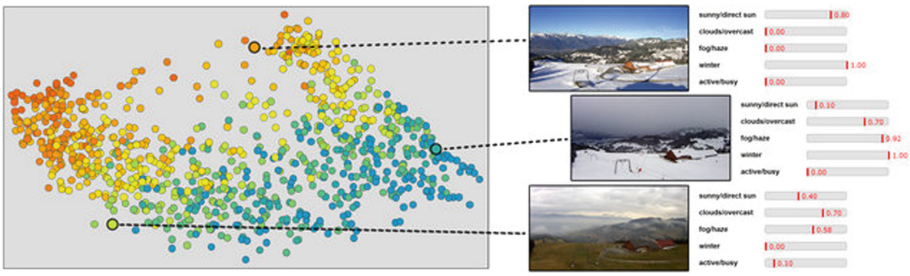


Fig. 8. Embeddings may also be colourised according to the state of background variables, revealing distributions in the latent space which can lead to better understandings and inference results.

5.3 Gallery Management

We propose that a function to select all embeddings for each class, delete old embeddings given there are more than N (an arbitrary number which may change based on performance results) embeddings for a class and then to compute and remove outliers by some method, e.g. Median Absolute Deviation (MAD) that the representativity of the gallery embeddings of the ground truth, and hence classification accuracy could be improved.

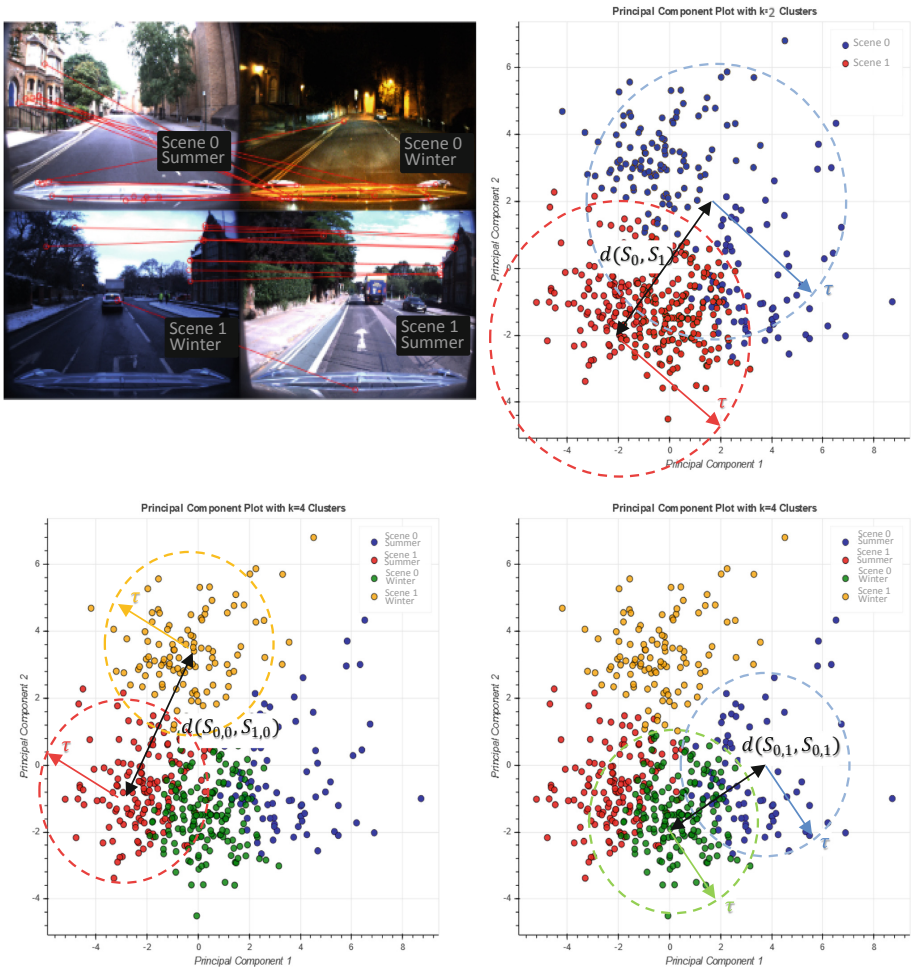


Fig. 9. A PCA projection of the latent space in DML showing how priori knowledge of background variables, e.g. seasonal variations in outdoor scenes in place recognition, may be used to minimize the intra-class variance and inter-class similarity such that the distance threshold, τ, is less than the distance between classes, d(S_0,0, S_1,0).

The embeddings are typically written into the HDF5 file in many of the GitHub repositories of previous work. This file format is useful for accessing large amounts of data quickly, however, it does not facilitate the removal of data entries as is desired, e.g. for removing old/noisy embeddings from the gallery over time.

Also, the integration of adaptive thresholding [47] or deep variational metric learning [48] which are methods which allow the distance threshold under which query embeddings must be from embeddings in the gallery to be classified variant to the distribution of embeddings could improve results even more substantially with our proposed method for gallery maintenance.

6 Conclusion

This paper investigates how the mapping element of DML may be exploited in situations where the salient features in arbitrary classification problems vary dependent on auxiliary background variables. Through the use of visualisation tools for observing the distribution of DML representations per each query variable for which prior information is available, the influence of each variable on the classification task may be better understood. Based on these relationships, prior information on these salient background variables may be exploited at the inference stage of the DML approach by using a clustering algorithm to improve classification performance. This research proposes such a methodology establishing the saliency of query background variables and formulating clustering algorithms for better separating latent-space representations at run-time. The paper also discusses online management strategies to preserve the quality and diversity of data and the representation of each class in the gallery of embeddings in the DML approach. We also discuss latent works towards understanding the relevance of underlying/multiple variables with DML.

6.1 Future Work

Performance comparison with existing not been achieved in this investigation work, however, the concept has promising future results, and the obvious next step in this investigation is to implement our approach on a publically available dataset to ensure reproducibility. The implementation of the proposed solution may be performed, for example, using the 3DWF dataset which contains demographic data such as age or gender is provided for every subject of a face dataset. By taking age, gender and ethnicity as the desired output variables in a multi-task metric learning approach primarily aimed at age estimation from 3D face data. We propose to project the discovered latent space to a representation with dimensions/directions for age, gender and ethnicity. In this way, we may demonstrate how our approach may be used to interpret relationships between binary, ordinal, continuous and seemingly nominal variables.

User interface could be the difference between powerful machine learning tools being a black box that may or not be trusted or a cognitive tool that extends human capabilities at understanding complicated data streams. Reasoning about data through representations can be useful even for kinds of data we understand well because it can make explicit and quantifiable things that are normally tacit and subjective. We propose that the latent space occupied by the representation discovered by metric learning may be exploited.

Acknowledgment. This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

1. Sanyal, S.: Discriminative descriptors for unconstrained face and object recognition (2017)
2. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification (2017)
3. Zheng, L., Yang, Y., Hauptmann, A.G.: Person re-identification: past, present and future (2016)
4. Wang, H., Li, H., Peng, J., Fu, X.: Multi-feature distance metric learning for non-rigid 3D shape retrieval. Multimed. Tools Appl. **78**, 30943–30958 (2019). https://doi.org/10.1007/s11042-019-7670-9
5. Boiarov, A., Tyantov, E.: Large scale landmark recognition via deep metric learning (2019). https://doi.org/10.1145/3357384.3357956
6. Bonadiman, D., Kumar, A., Mittal, A.: Large scale question paraphrase retrieval with smoothed deep metric learning (2019)
7. da Silva, A.C.M., Coelho, M.A.N., Neto, R.F.: A music classification model based on metric learning applied to MP3 audio files. Expert Syst. Appl. **144**, 113071 (2020). https://doi.org/10.1016/j.eswa.2019.113071
8. Thakur, A., Thapar, D., Rajan, P., Nigam, A.: Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss. J. Acoust. Soc. Am. **146**, 534–547 (2019). https://doi.org/10.1121/1.5118245
9. Marasović, T., Papić, V.: Accelerometer based gesture recognition system using distance metric learning for nearest neighbour classification. In: IEEE International Workshop on Machine Learning for Signal Processing, MLSP (2012)
10. Jeong, Y., Lee, S., Park, D., Park, K.H.: SS symmetry accurate age estimation using multi-task siamese network-based deep metric learning for front face images (2018). https://doi.org/10.3390/sym10090385
11. Rahman, S., Khan, S., Porikli, F.: Zero-shot object detection: learning to simultaneously recognize and localize novel concepts (2018)
12. Altae-Tran, H., Ramsundar, B., Pappu, A.S., Pande, V.: Low data drug discovery with one-shot learning. ACS Cent. Sci. **3**, 283–293 (2017). https://doi.org/10.1021/acscentsci.6b00367
13. Dong, N., Xing, E.P.: Domain adaption in one-shot learning (2018)
14. Rao, D.J., Mittal, S., Ritika, S.: Siamese neural networks for one-shot detection of railway track switches (2017)
15. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: International Conference on Learning Representations, pp. 1–11 (2017)
16. Fei-Fei, Li, Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 594–611 (2006). https://doi.org/10.1109/TPAMI.2006.79
17. Fe-Fei, L., Fergus, R., Perona, P.: A Bayesian approach to unsupervised one-shot learning of object categories. In: Proceedings Ninth IEEE International Conference on Computer Vision, vol. 2, pp.1134–1141. IEEE (2003)
18. Reed, S., Chen, Y., Paine, T., et al.: Few-shot autoregressive density estimation: towards learning to learn distributions. arXiv Prepr arXiv:171010304 (2018)
19. Mehrotra, A., Dukkipati, A.: Generative adversarial residual pairwise networks for one shot learning (2017)
20. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features (2016)
21. Pahde, F., Puscas, M., Wolff, J., et al.: Low-shot learning from imaginary 3D model (2019)
22. Santoro, A., Bartunov, S., Botvinick, M., et al.: One-shot learning with memory-augmented neural networks (2016)

23. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-SGD: learning to learn quickly for few-shot learning (2017)
24. Hou, S., Feng, Y., Wang, Z.: VegFru: a domain-specific dataset for fine-grained visual categorization (2017)
25. (PDF) Attention for Fine-Grained Categorization. https://www.researchgate.net/publication/269933088_Attention_for_Fine-Grained_Categorization. Accessed 16 Jan 2020
26. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization (2016)
27. Hansen, M.F., Smith, M.L., Smith, L.N., et al.: Towards on-farm pig face recognition using convolutional neural networks. Comput. Ind. **98**, 145–152 (2018). https://doi.org/10.1016/j.compind.2018.02.016
28. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification (2018)
29. Liao, W., Yang, M.Y., Zhan, N., Rosenhahn, B.: Triplet-based deep similarity learning for person re-identification (2017)
30. Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M.: Deep learning for IoT big data and streaming analytics: a survey (2017)
31. Gouk, H., Pfahringer, B., Cree, M.: Fast metric learning for deep neural networks (2016)
32. Surya Prasath, V.B., Alfeilat, H.A.A., Hassanat, A.B., et al.: Effects of distance measure choice on KNN classifier performance-a review (2019)
33. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition (2015)
34. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering (2015). https://doi.org/10.1109/cvpr.2015.7298682
35. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: Lecture Notes on Computer Science (Including Subser Lecture Notes on Artificial Intelligence Lecture Notes on Bioinformatics). LNCS, vol. 9910 (2016).. https://doi.org/10.1007/978-3-319-46466-4_29
36. Horiguchi, S., Ikami, D., Aizawa, K.: Significance of softmax - based features over metric learning - based features. In: ICLR 2017 (2017)
37. Huang, C., Loy, C.C., Tang, X.: Local similarity-aware deep feature embedding (2016)
38. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss (2016)
39. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification (2016)
40. Dong, X., Shen, J., Wu, D., et al.: Quadruplet network with one-shot learning for fast visual object tracking (2019)
41. Sohn, K.: Improved deep metric learning with multi-class N-pair loss objective (2016)
42. Zhou, M., Niu, Z., Wang, L., et al.: Ladder loss for coherent visual-semantic embedding (2019)
43. Zhang, Y., Yang, Q.: A Survey on multi-task learning (2017)
44. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009). https://doi.org/10.1145/1577069.1577078
45. Bonaccorso, G.: Machine Learning Algorithms Popular Algorithms for Data Science and Machine Learning, 2nd edn. Packt Publishing Ltd, Birmingham (2018)
46. Larsson, M., Stenborg, E., Hammarstrand, L., et al.: A cross-season correspondence dataset for robust semantic segmentation (2019)
47. Wu, L., Wang, Y., Gao, J., Li, X.: Deep adaptive feature embedding with local sample distributions for person re-identification (2017)
48. Lin, X., Duan, Y., Dong, Q., et al.: Deep variational metric learning (2019)

An Automated System of Threat Propagation Using a Horizon of Events Model

Kilian Vasnier[1,2(✉)], Abdel-Illah Mouaddib[1], Sylvain Gatepaille[2], and Stéphan Brunessaux[2]

[1] Université de Caen - GREYC,
Boulevard du Maréchal Juin, 14000 Caen, Calvados, France
kilian.vasnier@unicaen.fr
[2] Airbus Defense and Space,
1 Boulevard Jean Moulin, 78990 Élancourt, Yvelines, France

Abstract. Situation Awareness area deals with an ever-growing amount of data to be processed. Decision makers need new tools to swiftly assess a situation, in spite of the huge amount of information to interpret. This reality is even truer in Crisis Analysis such as military and rescue domains. Considering the speed with which information is acquired, it is crucial to propose an efficient decision process. Automated systems are absolutely necessary for decision makers as they enable them to save a valuable time while making decisions quickly in order to solve problems. In this paper, we present an improvement of an automated threat propagation model through a dynamic environment. This model proposes a heat map of the potential threat of an enemy attacking different points. For a human operator, knowing the path and the threat in the upcoming moments rather than the final objective of the enemy is crucial to deploy his sensors or even establish early counter-measures. This work takes place in a military use case based on the NATO military doctrine. The proposed scenario describes an army attacking the borders of a country. This enemy tries to make the most of its troops while they are breaking through the defensive lines. Intelligence services identify specific zones to observe and understand the enemy's intentions. The model provides a topographical structure that can be read like a graph structure representing the different observable zones that can be observed along with their potential targets.

Keywords: Situation awareness · Crisis analysis · Threat propagation · Dynamic environment modelling

1 Introduction

In many areas, it is a great challenge to solve difficult situations (e.g. tactical or rescue missions) while having to assess more and more data. The amount of information to process is rising and thus, human experts face difficulties to deal

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 114–131, 2021.
https://doi.org/10.1007/978-3-030-55180-3_9

with this considerable volume of data. Situation Awareness (SA), depicted in [5] and [6], offers solutions to help in decision-making issues. Furthermore, experts will have to take a massive quantity of parameters into account before making a decision.

A more specific part of the SA, which is the crisis situation analysis [9,13], brings strong time related constraints. As the human operator must make quick and critical decisions despite the limited information retrieval resources, the purpose is to present a reliable representation of the situation. Here, the speed of acquisition and processing of information must be coupled with an analysis as similar as possible to the reality.

As these problematics deal with dynamic and complex environments, modelling is a hard task to achieve. So the representation of the situation is a crucial point as the quality of the assessment relies on the quality of the modelling. This is all the more important as the computational constraints for automated solutions need a clear and easy way to handle the representation of information.

These problems are related to notable works in Game theory with different approaches such as [1,11] with min/max learning adversarial strategy. More recently, another work presents a new alpha/beta function [8] and [3] dealing with four warfare scenarios. Other works discuss similar problems in stochastic and partially observable environments [12] and more specifically in warfare scenarios [15]. The maritime field benefits from several works where the automated systems are more and more required as presented in [14].

The specific domain of warfare information suits both Game Theory problems and crisis situation analysis. A study from a workshop [2] greatly sums up the challenges and issues that need to be assessed in such a field. It also provides a complete survey [4] which presents numerous use cases, models and information representation problems in addition.

Nevertheless, the threat propagation is not assessed in these related works and in the warfare information field in general (aside from network information systems as in [16]). The following paper tends to propose an automated system to monitor the threat on a heat map to assess a battlefield situation.

This paper presents an improvement of an automated threat propagation model, applied on a scenario created from the military doctrine [7,10]. The claim is to propose a precise monitoring of the threat in a battlefield situation by (A) knowing the most probable paths and targets of the enemy and (B) knowing the required number of time steps for the enemy to reach each point and target.

2 Definition of the Problem

The scenario presents an attack from a country A against another country B. The attacker tries to break through the borders of the defender using different possible paths.

The intelligence services characterize the High Level Priority Information Zones (HPIZ) on the battlefield which correspond to the potential crossing points

of enemy troops. The available defender's sensors have to collect the information from these different zones to understand the target of the invader and consequently its strategy to succeed in its assault.

Here, the strategy of the enemy is to find which area he will assault. These different zones, named Attack Points (AP), must be protected and the decision maker has to promptly understand how to dispatch its troops to optimize its defence.

To assess the attack plan, the operational system must detect and depict the evolution of the threat during all the different attack phases. The threat corresponds to the way the enemy moves its companies. Companies are a clustering of military units with a specific role and force of destruction. The higher the force of a company is, the higher its threat potential is.

The problem focuses on battlefield zones crisscrossed with paths and routes. Consequently, the best representation is a graph. Considering a Directed Acyclic Graph (DAG) $G = V, E$ with V the set of vertices and E the set of the edges, which represents a topographical graph (see Fig. 1).

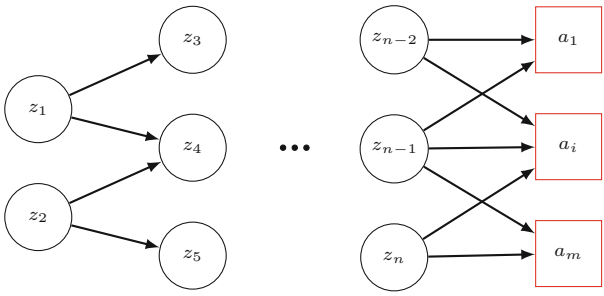


Fig. 1. Sample of topographical graph

Let $Z = \{z_1, ..., z_n\}$ a set of HPIZ as $Z \subseteq V$ and $\forall z \in Z, \pi(z) \neq \emptyset$ with $\pi(z)$ the children of z representing the set of accessible zones from z.

Considering $A = \{a_1, ..., a_m\}$ the set of AP such as $A = V \backslash Z$ and $\forall a \in A, \pi(a) = \emptyset$, as they are the leaves of the graph, corresponding to all potential targets of the enemy.

Given E the set of edges as $\forall e \in E$, $e^{(u \rightarrow v)}$ is a path between the nodes $u, v \in V$ depicting the possibility for troops to go from u to v. Each edge e has a weight noted $w_e = [0, 1] \in \mathbb{R}$ corresponding to the probability for a company to go through this path. The sum of all the outgoing edges from a node v, denoted $w_{e^{z^+}}$, is equal to 1.

Each edge also possesses a type T_e characterizing the difficulty to cross a specific path. The type defines if the path goes through a forest, is a road, a mountain, a river and so on.

Phases of the attack are stated as follows: $P = \{p_1, ..., p_k\}$ a set of phases with p defined as $p \subseteq V$ knowing $\forall p \in P \backslash p_k \subseteq Z$. The phase p_1 corresponds to

all the root nodes as $\forall z \in Z, u(z) = \emptyset$ with $u(z)$ the parents node of z. Phases p_2 up to p_{k-1} are defined by $p_i = \bigcup_{z \in p_{i-1}} \pi^*(z)$ with $\pi^*(z) = \pi(z) \backslash z \in p_{i-1}$. And finally, $p_k = \{A\}$ containing all the AP nodes.

The enemy's troops are stated as $C = \{c_1, ..., c_q\}$ the set of companies. Each company is described by a type T_c (infantry, tank, artillery, ...). The type of a company involves firstly a threat score $TS(c) \in \mathbb{N}$ representing its role and its destruction force and secondly, a crossing index $Cr(c) = [0, 1] \in \mathbb{R}$ depending on the type of the path T_e it goes through. A crossing index equals 0 represents the impossibility to go through the corresponding path.

All companies present in a HPIZ z at time step t are denoted C_z^t which is an empty set if no company is in it. The threat score at time step t of a zone is defined as:

$$M(z_i)^t = \sum_{c_k \in C_{z_i}^t} TS(c_k) \tag{1}$$

The transition function from zone z_i to zone z_j is given by the transition matrix $T(c_k \in C_{z_i}^t, \pi(z_i))$ which returns all transition probabilities, denoted $P(c^{z_i \rightarrow z_j})$, from z_i to one of its children with $z_j \in \pi(z_i)$. The probability of transition is defined as follows:

$$P(c^{z_i \rightarrow z_j}) = w_e^{(z_i \rightarrow z_j)} \cdot Cr(c) \cdot TS(c) \tag{2}$$

3 Automated Threat Propagation Model

3.1 A Pessimistic Approach

The threat propagation considered in this work is based on a pessimistic method. This method acknowledges the worst case of an attack, namely a scenario with an optimal attack causing the worst damages to the defender.

The ensuing hypothesis is to consider the enemy will still move forward and try to optimize the path of the attack with its companies to be the most efficient and achieve a quick-fire attack.

The idea is to compute the threat generated by each company for all its possible moves and propagate it. A world is generated from the combinatory of all possible non-concurrent moves of all observed enemy's companies. A world is the representation of the environment (as defined in Sect. 2) with the propagation of the threat on each zone (HPIZ and AP). A non-concurrent movement is considered as all the possible combinations of each company whenever a company is present in only one accessible zone in a resulting potential world. Then, the system compares the global threat of each potential strategy and the highest scores are considered as the most plausible scenarios the enemy will adopt.

3.2 Formal Definition of the Model

The propagation starts from the root nodes as $\forall z \in Z$ given $u(z) = \emptyset$. The propagation goes through each child $\pi(z)$ for each phase up to the AP.

The threat propagation is computed after a potential movement for each company at $t + 1$ to estimate the highest threat it can generate. But as a company could behave irrationally (i.e. not optimally as possible), the model proposes a set of best possible strategies, the k-best strategies, to assess several possibilities with a strong probability.

The propagation is thus done at $t + 1$ and the resulting threat propagation from a HPIZ z_i to another zone (HPIZ or AP) z_j, denoted $Prop(z_i)^{t+1}$, is then computed as follows:

$$Prop(z_i)^{t+1} = P(z_i \rightarrow z_j) \cdot \overline{Cr(C_{z_i}^{t+1})} \tag{3}$$

With $P(z_i \rightarrow z_j)$ the probability to take the path from z_i to z_j. So the total resulting threat score of z_j at time step $t + 1$ is therefore computed as:

$$M(z_j)^{t+1} = prop(z_i)^{t+1} + M(z_j)^t \tag{4}$$

Which represents the companies movement to z_j with $M(z_j)^t$ the current threat score of z_j. The threat takes into account the probability of companies to go from z_i to z_j and also the means of the crossing indexes of all companies included in the moves of this specific world as well as the global threat of companies like:

$$\overline{Cr(C_{z_i}^{t+1})} = \frac{\sum_{c_k \in C_{z_i}^{t+1} \cup Cz... \rightarrow {z_i}^{t+1}} \Big(TS(c_k) \cdot Cr(c_k) \Big)}{|C_{z_i}^{t+1} \cup Cz... \rightarrow {z_i}^{t+1}|} \tag{5}$$

With $Cz... \rightarrow {z_i}^{t+1}$ as all the companies in all the nodes after the movement which could reach the node z_i from the root. This includes all troops which can reach z_i from its parents $u(z_i)$ and their parents up to the root.

Formally, the threat propagation is represented by the following formula:

$$M(c_k, z_j) = TS(c_k) * P(z_c \rightarrow z_j) * Cr(c_k^{z_c \rightarrow z_j}) \tag{6}$$

Where z_c is the HPIZ where the company is at time step $t + 1$ and $z_c \rightarrow z_j$ represents all the HPIZ crossed from z_c to z_j. This score is computed for each company for each zone it can reach.

To avoid irrational potential behaviours from companies, the propagation considers a drudgery or fatigue score. If this parameter is not taken into account within the model, companies tend to cross difficult grounds without penalty since the reward for crossing this environment leads to a better threat score. So the threat of the companies at $t + 1$ is represented by:

$$M(c_k^{t+1}) = TS(c_k) * Cr(z_i^t \rightarrow z_i^{+1}) \tag{7}$$

With z_i the position of c_k at t and z_i^{+1} the position at $t+1$. This addition does not prevent a company from crossing a laborious path if the resulting potential threat is better than another and is more representative of rational behaviours.

Each combination of potential moves is consequently propagated for each company, and for each non-concurrent combinatory of all companies, a potential world Φ at time step $t+1$ is generated where the total threat is calculated as:

$$M(\Phi) = \sum_{a_i \in \Phi} M(a_i) \tag{8}$$

This resulting score allows to compare which world Φ generates the highest threat at the very last moment (implying the moment when the companies reach the AP). From these potential worlds, the system can assess which strategy is most likely to happen and propose the k-best strategy sorted out in a descending order to highlight only the best ones.

3.3 Algorithms

The implementation requires to compute a combinatory of non-concurrent combinations from each company (i.e. combinations that can exist in a same world at a same time step t to avoid a company to be present in several zones). However, even if the complexity increases with the number of accessible zones and the number of companies, in all studied use cases, these numbers never reach an amount of zones and companies preventing the possibility of generating all potential worlds.

The generation of potential non-concurrent worlds are generated as shown in Algorithm 1. The number of worlds returned corresponds to the considered k-best strategies. The result is sorted in decreasing order in keeping with the highest threat score given by Eq. 8.

Algorithm 1. Generation of all potential worlds

```
PotentialWorlds ← ∅
for each HPIZ z ∈ Z do
    Combinations ← generateAllMovementCombinations()
    for each combination comb ∈ C do
        potentialWorld ← generateWorldFrom(comb)
        propagateThreat() {see Algorithm 2}
        PotentialWorlds ← PotentialWorlds ∪ potentialWorld
    end for
end for
NonConcurrentWorlds ← combineNonConcurrentWorldsFrom(PotentialWorlds)
sortByHighestThreatScore(NonConcurrentWorlds)
return NonConcurrentWorlds
```

In application, the threat propagation is proceeded as shown in Algorithm 2. As the threat is propagated from the root node up to the AP, and only one node

with companies is considered in one possible world, the function only takes the parent node threat and propagates it to the child (flooding).

Algorithm 2. Threat Propagation

```
1: for each phase p ∈ P\p_k do
2:   for each HPIZ z ∈ p do
3:     if z is a root node then
4:       threat ← sumCompaniesThreat() {see Eq. 1}
5:       z.threatScore ← threat
6:     end if
7:     for each children child ∈ π(z) do
8:       threat ← propagateThreatFrom(z) {see Eq. 6}
9:       child.threatScore ← threat
10:    end for
11:  end for
12: end for
```

4 Threat at a Specific Horizon

Once a heat map of threat propagation is available to monitor a situation, a human operator also needs to understand when the attack will happen. To do so, experts need to understand how the total threat of each point will diffuse through next the time steps until all the enemy troops arrive on their targets.

Considering H as the maximal horizon of the events (here, the attack on one or several AP). Let h be the horizon of the events at a specific time where we want to know the potential propagation of the threat.

Given $Vel(c)$ the velocity of the company c representing the distance c can travel in one time iteration, and $Dist(z_i \rightarrow z_j)$ the distance between zones z_i and z_j based on the same unit of $Vel(c)$, we note $Zones(c)^t + h$ the set of zones that the company c can reach at horizon h. Inversely, all companies that could reach a zone z are denoted as $Comp(z)^{t+h}$ at specific time $t + h$.

Figure 2 shows a short example with $Vel(c) = 1$ and for each zone z_i and its children z_j, $Dist(z_i \rightarrow z_j) = 1$. The company standing in z_1 can reach all the zones at horizon H such as $Zones(c)^{t+H} = [z_2, z_3, a_1, a_2]$ and for $h = 1$, $Zones(c)^{t+h} = [z_2, z_3]$.

The threat propagation can thus be defined as:

$$M(c_k)^{t+h} = M(c_k^{t+1}) \cdot \epsilon \tag{9}$$

$$\epsilon = \begin{cases} 1, \text{if } Dist(z_c \rightarrow z_j) \leq Vel(c) \cdot h \\ 0, \text{otherwise} \end{cases} \tag{10}$$

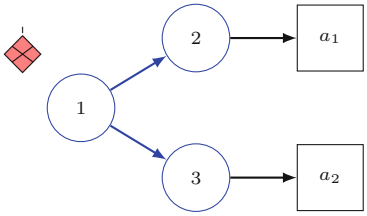


Fig. 2. Theorical example

With $M(c_k^{t+1})$ from the Eq. 7 and $Dist(z_c \rightarrow z_j)$ the set of the edges crossed from the zone z_c where the company c stands up to the zone z_j as:

$$Dist(z_c \rightarrow z_j) = \sum_{d \in z_c \rightarrow z_i \cup \ldots \cup z_{j-1} \rightarrow z_j} Dist(d) \tag{11}$$

In cases where several paths lead to z_j from z_c, the only distance we are interested in is the Shortest Path as the purpose is to know if a zone is reachable by the company and if the threat has to be propagated or not.

This concept could be summed up to estimate how many steps are required for a company observed in a specific zone to reach each zone. To that end, an equivalent of the Dijkstra algorithm is used.

Figure 3 illustrates this with an example at $t = 1$ where a company stands in zone z_1. An interesting aspect here is to consider the possibility for c to reach zone z_4 as $Zones(c)^{t+1} = [z_2, z_3, z_4]$. For a human operator, such information could be crucial to understand how to deploy his sensors in order to assess more quickly the enemy strategy.

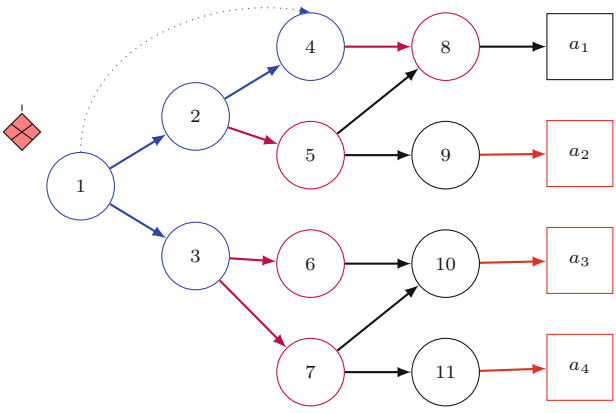


Fig. 3. Example at $t = 1$ with a company in z_1

This improvement also allows to know how the threat will diffuse through time. As the maximum threat is known (as shown in Sect. 3), the threat at a specific horizon permits to know when a company could arrive at minimum time in a zone and consequently its threat score.

So the threat of a zone is represented by a set of threat scores as $M(z) = [M(z)^t, M(z)^{t+1}, ..., M(z)^{t+H}]$ with H the last horizon of events (i.e. the maximal threat score computed by Eq. 7). In addition to this, a potential world Φ has a corresponding time describing the number of steps necessary to arrive in such a situation giving more or less priority to deal with them. As an example, in the world Φ in Fig. 3, a_1 is threatened by the company at $h = 3$ and a_2, a_3, a_4 at $h = 4$.

Each threat score $M(z)^{t+i}$ (with $1 \geq i \geq H$) of each zone is then computed as:

$$M(z)^{t+i} = \sum_{c_k \in Comp(z)^{t+i}} M(c_k^{t+1}) \tag{12}$$

With this system of information, a human operator is able to focus on most critical scenarios while taking into account both the threat and the urgency of the situation, withe the knowledge of the remaining time (left over time) before the scenario happens. This represents a strong added value as observing all the zones is not possible because observing resources is limited in a crisis situation. With time and threat information, the decision to deploy sensors is far more informed and efficient to assess the situation in the next moment.

5 Experiments and Results

We present within our results three simulated scenarios based on the same topographical environment (see Fig. 4) with their 5-best strategies.

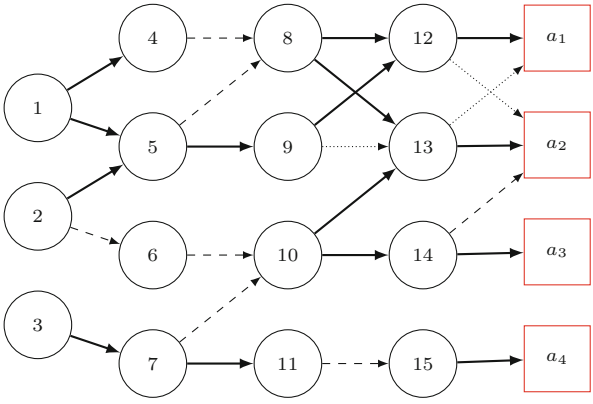


Fig. 4. Topological graph for scenarios

In these scenarios are considered three types of companies; infantry, tank and artillery. The associated threat score for each type are shown in Table 1 (companies' symbol respects the APP-6 standards). The paths of the map are described by three types, namely A, B and C (they can be considered respectively as roads, forests and rivers) with the corresponding crossing index for each companies' type presented in Table 2. Straight edges represent the type A, dashed edges the type B and dotted edges the type C. All these data are purely artificial to demonstrate the performance of the proposed threat propagation model described in Sects. 3 and 4. These indexes are defined by an expert knowledge in the military domain by their role and movement capacity in an operational case.

Table 1. TS(c)

Company Type	Threat Score
Infantry	3
Tank	5
Artillery	1

Table 2. Cr(c)

Company Type	Path Type		
	A	B	C
Infantry	1.0	0.8	0.2
Tank	1.0	0.6	0.1
Artillery	1.0	0.4	0.05

The three scenarios are described step by step in the Table 3 to enable experiment repetition. Their respective results are presented both in a Fig. 6, 8, 10 and a Tables 4, 5, 6 in the following subsections. Figure 5 shows an example of the heat map of the threat during a k-best strategy (here the 1-best strategy of the scenario 1 at step 2).

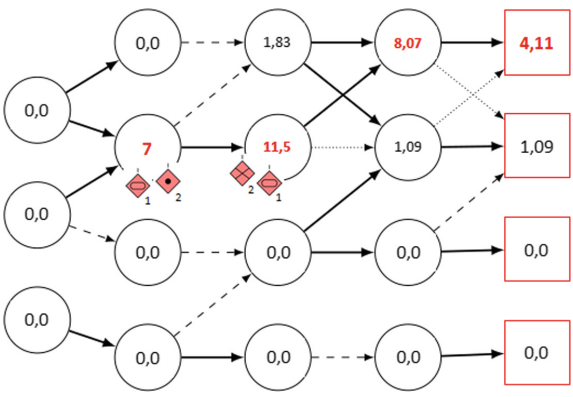


Fig. 5. Example of heat map for scenario 1 (step 2)

Table 3. Scenarios

Company Type	t_1	t_2	t_3	t_4	t_5	t_6
Infantry	$1 : z_1$ $1 : z_2$	$2 : z_5$	$2 : z_9$	$2 : z_{12}$	$2 : z_{12}$	$2 : a_1$
Tank	$1 : z_1$ $1 : z_2$	$1 : z_2$ $1 : z_5$	$1 : z_9$ $1 : z_5$	$1 : z_{12}$ $1 : z_9$	$2 : z_{12}$	$2 : a_1$
Artillery	$\emptyset$	$2 : z_2$	$1 : z_5$ $1 : z_2$	$1 : z_9$ $1 : z_5$	$2 : z_9$	$2 : z_{12}$

(a) Scenario 1 - optimized attack

Company Type	t_1	t_2	t_3	t_4	t_5	t_6
Infantry	$2 : z_1$ $1 : z_3$	$2 : z_5$ $1 : z_7$	$2 : z_9$ $1 : z_{10}$	$2 : z_{12}$ $1 : z_{14}$	$2 : z_{12}$ $1 : z_{14}$	$2 : a_1$ $1 : a_3$
Tank	$1 : z_3$	$2 : z_1$ $1 : z_3$ $1 : z_7$	$2 : z_5$ $1 : z_7$ $1 : z_{10}$	$2 : z_9$ $1 : z_{10}$ $1 : z_{14}$	$2 : z_{12}$ $2 : z_{14}$	$2 : a_1$ $2 : a_3$
Artillery	$\emptyset$	$1 : z_3$	$2 : z_1$ $1 : z_3$ $1 : z_7$	$2 : z_5$ $2 : z_7$	$2 : z_9$ $2 : z_{10}$	$2 : z_{12}$ $2 : z_{14}$

(b) Scenario 2 - multiple-objectives

Company Type	t_1	t_2	t_3	t_4	t_5	t_6
Infantry	$2 : z_2$ $1 : z_3$	$2 : z_6$ $1 : z_7$	$3 : z_{10}$	$3 : z_{13}$	$3 : z_{13}$	$3 : a_2$
Tank	$\emptyset$	$2 : z_2$ $1 : z_3$	$2 : z_6$ $1 : z_7$	$3 : z_{10}$	$3 : z_{13}$	$3 : a_2$
Artillery	$\emptyset$	$1 : z_3$	$1 : z_2$ $2 : z_3$ $1 : z_7$	$1 : z_6$ $3 : z_7$	$1 : z_7$ $3 : z_{10}$	$4 : z_{10}$

(c) Scenario 3 - deceived strategy

5.1 First Scenario - Proof of Concept

The first scenario acts as a proof of concept to demonstrate the efficiency of the threat propagation model as it reveals the enemy's strategy. First observed companies start at HPIZ z_1 and z_2 and go through the most optimal and efficient path (z_5, z_9, z_{12}) to attack the AP a_1. The assaulter strikes only one target and concentrates all his forces to achieve a quick-fire attack.

Figure 6 shows clearly a_1 as the only target as early as the time step 2 and it is confirmed by all the 5-best strategies and the following iteration.

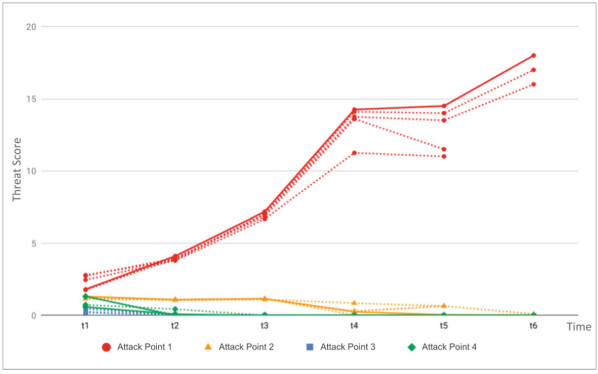


Fig. 6. Results of 5-best strategies for scenario 1

Figure 7 concerns the 1-best strategy at time step 2. This specific time step is interesting to analyse as it is the moment where the enemy strategy begins to be inferred correctly. Thanks to this, the operator can understand that the attack on a_1 has good chances to occur in 3 time iterations ($h = 3$) even if the maximum threat occurs on $h = 5$. The main interest is to understand that it is not valuable to observe the AP a_1 before $h = 3$ and to focus on observing the previous zones, in this case z_{12} or z_{13}, to confirm the inference.

This scenario confirms the ability of the system to infer correctly an optimal attack in sufficient time and permits to optimized both the observation between each time step and the defense to set up as a counter-measure.

5.2 Second Scenario - Multi-objectives Attack

Second scenario shows the ability to monitor a multi-objectives attack on multiple points. In this specific situation, the targets are a_1 and a_2. To do so, the attacker will focus on the line z_5, z_9, z_{12} as before and the line z_6, z_{10}, z_{13} from the HPIZ z_1 and z_2.

In this scenario, we can notice there is not always 5 strategies as for the first time step. During the two first time steps, the enemy's intention is not clearly identify for the reason that a_4 is still shown as a potential target, but the threat score of a_1 raise substantially. The reliability of the strategy starts to be reveal on the time step 4 where the threat score of a_2 and a_4 decreased dramatically while those of a_1 and a_3 raise. Step 4 thus confirms the multi-targets attack with a very high threat score on both a_1 and a_3. All the 5-best strategies depicts the same situation and steps 5 and 6 (the attack) confirm this strategy.

Table 4. Threat score of attack points - scenario 1

AP	k-best	t_1	t_2	t_3	t_4	t_5	t_6
a_1	1-best	1,81	4,11	7,18	14,25	14,5	18
	2-best	1,77	3,96	7,01	14,1	14	17
	3-best	2,79	3,83	7,03	13,75	13,5	16
	4-best	2,75	3,96	6,86	13,6	11,5	–
	5-best	2,47	3,81	6,68	11,25	11	–
a_2	1-best	1,31	1,09	1,16	0,25	0,05	0
	2-best	1,3	1,1	1,17	0,02	0,05	0,05
	3-best	1,17	1,12	1,13	0,3	0,65	0,1
	4-best	1,15	1,06	1,14	0,28	0,65	–
	5-best	1,15	1,07	1,1	0,85	0,65	–
a_3	1-best	0,47	0	0	0	0	0
	2-best	0,47	0,02	0,02	0	0	0
	3-best	0,21	0,05	0	0	0	0
	4-best	0,21	0	0,02	0	0	–
	5-best	0,26	0,02	0	0	0	–
a_4	1-best	1,32	0	0	0	0	0
	2-best	1,32	0,04	0,03	0	0	0
	3-best	0,58	0,087	0	0	0	0
	4-best	0,58	0	0,03	0	0	–
	5-best	0,73	0,43	0	0	0	–

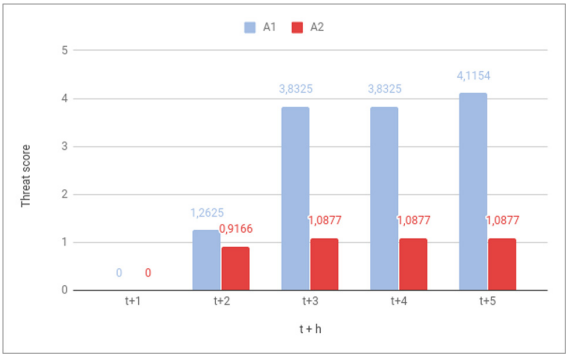


Fig. 7. Horizon of events of 1-best strategies at time step 2 for scenario 1

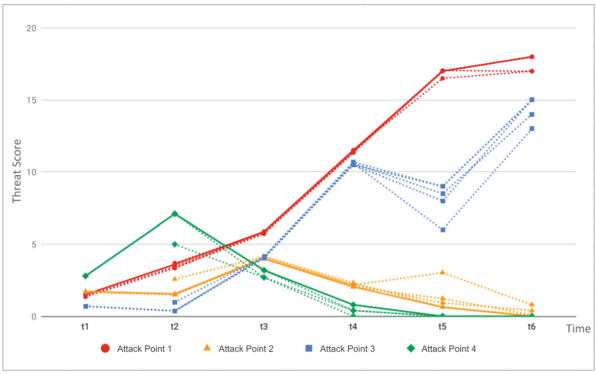


Fig. 8. Results of 5-best strategies for scenario 2

Figure 9 concerns the 1-best strategy at time step 3 where all AP seems to be potential targets. The histogram shows the AP a_1 is very threatened in a short time, $h = 2$. For all other AP, the delta between all threat scores is not enough reliable but the moment of the attack should be pretty close.

The human expert could consequently decide which zone must be observed to discriminate which AP will be attacked as a_2, a_3 and a_4 are sufficiently threatened to consider them as potential targets while he already know that a_1 will be attacked in the time step 2.

This scenario confirms the automated threat propagation system is able to show multi-objectives attack quite in a good time with enough confidence.

5.3 Third Scenario - A Deceiving Strategy

The last scenario allows us to evaluate the limits of the model as its current formalization with a deceiving strategy of the attacker. This time, the enemy starts from HPIZ z_2 and z_3 and go through line z_6, z_{10}, z_{13} and the line z_7, z_{10}, z_{13} and behaves as it will attack AP a_3 but changes at the last moment to go on a_2 to fake its strategy.

The presented scenario is more complex to infer as the enemy seems to follow a specific path towards a_3 and/or a_2 and ultimately ending to only attack a_2. Even if the system presents a_2 as the main target at the step 4, the real intention of the enemy is shown clearly at time step 5 as the delta between a_2 and a_3 sufficiently increase to understand that a_2 will be the only target of the attack.

Like the second scenario, Fig. 11 (based on the 5-best strategy at time step 3) depicts a likely attack in the two next iterations. As the AP a_2 and a_3 are the most threatened, the next zone to observe are definitely z_{13} and z_{14}. Besides, the horizon of events of these two zones indicates a very likely movements of enemy troops for the next iteration and knowing if troops will arrive in z_{13} or z_{14} could allow to know with a better confidence which is the target of the attack.

Table 5. Threat score of attack points - scenario 2

AP	k-best	t_1	t_2	t_3	t_4	t_5	t_6
a_1	1-best	1,51	3,61	5,88	11,5	17	18
	2-best	1,43	3,45	5,8	11,51	17,02	18
	3-best	1,35	3,35	5,73	11,3	5 17,05	17
	4-best	–	3,35	5,88	11,51	16,5	17
	5-best	–	3,7	5,81	11,35	17	18
a_2	1-best	1,73	1,56	4,01	2,05	0,63	0
	2-best	1,7	1,5	3,99	2,19	0,93	0,4
	3-best	1,67	1,51	3,99	2,03	1,23	0,1
	4-best	–	1,51	4,15	2,33	0,65	0,05
	5-best	–	2,58	4,15	2,17	3,03	0,8
a_3	1-best	0,7	0,37	4,05	10,5	9	15
	2-best	0,7	0,37	4,05	10,6	8,5	14
	3-best	0,7	0,37	4,05	10,5	8	15
	4-best	–	0,37	4,15	10,7	9	15
	5-best	–	0,97	4,15	10,6	6	13
a_4	1-best	2,8	7,1	3,2	0,8	0	0
	2-best	2,8	7,1	3,2	0,4	0	0
	3-best	2,8	7,1	3,2	0,8	0	0
	4-best	–	7,1	2,7	0	0	0
	5-best	–	5	2,7	0,4	0	0

Fig. 9. Horizon of events of 1-best strategies at time step 3 for scenario 2

In this scenario, the model understands the enemy strategy with a poorer result in time. Even if the target is confirmed during step 5, the threat score of AP could lead to misconception of the enemy strategy.

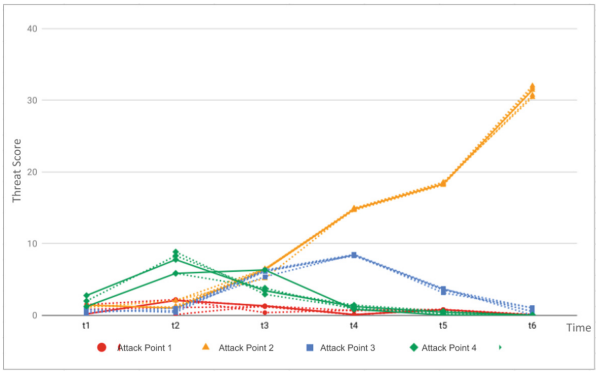


Fig. 10. Results of 5-best strategies for scenario 3

Table 6. Threat score of attack points - scenario 3

AP	k-best	t_1	t_2	t_3	t_4	t_5	t_6
a_1	1-best	0,16	2,05	1,29	0,06	0,75	0
	2-best	0,81	1,06	1,16	0,06	0,75	0,025
	3-best	1,46	2,17	1,21	0,07	0,76	0,05
	4-best	–	2,17	0,34	0,69	0,78	0,05
	5-best	–	0,07	1,29	0,61	0,75	0,075
a_2	1-best	1,40	0,96	6,31	14,77	18,27	31,4
	2-best	1,24	1,12	6,33	14,62	18,13	31,7
	3-best	1,08	2,04	5,11	14,91	18,41	32
	4-best	–	2,04	6,45	14,91	18,54	30,4
	5-best	–	1,28	6,46	14,76	18,27	30,7
a_3	1-best	0,78	0,37	6,1	8,35	3,6	1
	2-best	0,54	0,6	6,12	8,25	3,5	0,5
	3-best	0,3	0,97	5,35	8,45	3,7	0
	4-best	–	0,97	6,28	8,45	3,1	1
	5-best	–	0,82	6,2	8,35	3,6	0,5
a_4	1-best	2,74	7,74	3,4	1,2	0,4	0
	2-best	1,97	8,28	3,45	1,4	0,56	0
	3-best	1,2	5,82	6,3	0,8	0	0
	4-best	–	5,82	3,77	0,8	0,4	0
	5-best	–	8,82	2,9	1	0,16	0

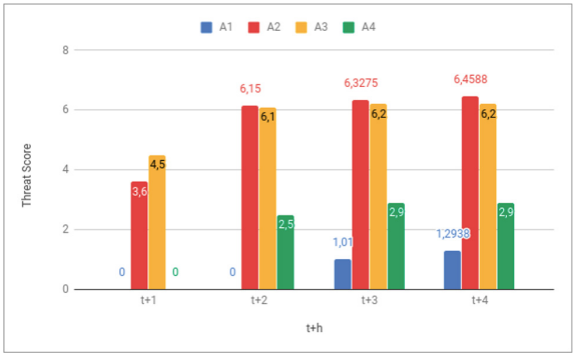


Fig. 11. Horizon of events of 5-best strategies at time step 3 for scenario 3

6 Conclusion

This paper introduces an improved automated threat propagation model applied to the warfare domain in a topological structure. The presented tool allows a human operator to monitor a battlefield situation and to understand how the enemy behaves. The enemy strategies are properly inferred by showing the vulnerability of each target and the time before an attack happens. The horizon of the events allows to easily understands what are the next zones to observe to discriminate more efficiently the enemy's strategy for the next steps.

Yet, two types of limits can be identified. A deceiving strategy slows substantially the inference and can lead to a wrong target in first time steps. The k-best strategy might not be the best heuristic to understand fake behaviour from the enemy troops.

Future works will focus on the possibility to represent defence systems on target that might be known or not by the enemy with a probability. This probability could direct the intention of an enemy to prefer a target rather than another one.

Another interesting perspective is the possibility to allow the human operator to add information about a company we are not able to observe at a specific time. For example, the decision maker can imagine the presence of other troops, such as a military engineering company which enables tanks to go through a river. Such a feature adds the possibility to anticipate a potential strategy asserted by an expert as a very potential strategy of the assaulter.

Acknowledgment. The authors acknowledge that this work is partially funded by the French MoD (DGA) in the framework of CIFRE-Defense contract no 005/2016/DGA.

References

1. Tridgell, A., Weaver, L., Baxter, J.: KnightCap: a chess program that learns by combining TD (lambda) with game-tree search. arXiv preprint cs/9901002 (1999)
2. Miller, W.L., Ott, A., Saydjari, O.S., Hamilton, S.N.: Challenges in applying game theory to the domain of information warfare. In: Information Survivability Workshop (2002)
3. Molsa, J.V., Jormakka, J.: Modelling information warfare as a game. J. Inform. Warfare **4**(2), 12–25 (2005)
4. Shafi, K., Merrick, K., Hardhienata, M.: A survey of game theoretic approaches to modelling decision-making in information warfare scenarios. Future Internet **8**(3), 34 (2016)
5. Mica, R., Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Hum. Factors **37**(1), 32–64 (1995)
6. Niklasson, L.: Extending the scope of situation analysis. In: 2008 11th International Conference on Information Fusion, pp. 1–8. IEEE (2008)
7. CEERAT PFT RENS: Manuel de l'unité de renseignement de brigade. Ministère de la défense (2016)
8. Schaeffer, J., Pijls, W., De Bruin, A., Plaat, A.: Exploiting graph properties of game trees. In: AAAI/IAAI1, pp. 234–239 (2002)
9. Quarantelli, E.L.: Disaster crisis management: a summary of research findings. J. Manag. Stud. **25**(4), 373–385 (1998)
10. CEERAT RENS: Manuel d'emploi du sous-groupement recherche multicapteurs. Ministère de la défense (2016)
11. Rivest, R.L.: Game tree searching by min/max approximation. Artif. Intell. **34**(1), 77–96 (1987)
12. Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., Wu, Q., Roy, S.: A survey of game theory as applied to network security. In: 43rd Hawaii International Conference on System Sciences, pp. 1–10 (2010)
13. Scott, P., Rogova, G.: Crisis management in a data fusion synthetic task environment. In: Procedings of FUSION (2004)
14. Vasnier, K., Mouaddib, A.-I., Gatepaille, S., Brunessaux, S.: Multi-level information fusion and active perception framework: towards a military application. NATO SET-262 RSM on Artificial Intelligence for Military Multisensor Fusion Engines (2018)
15. Vasnier, K., Mouaddib, A.-I., Gatepaille, S., Brunessaux, S.: Multi-level information fusion approach with dynamic bayesian networks for an active perception of the environment. In 2018 21st International Conference on Information Fusion (FUSION), pp. 1844–1850. IEEE (2018)
16. Xiaobin, T., Yong, Z., Hongsheng, X., Xiaolin, C.: A markov game theory-based risk assessment model for network information system. In: International Conference on Computer Science and Software Engineering, vol. 3, pp. 1057–1061 (2008)

Realizing Macro Based Technique for Behavioral Attestation on Remote Platform

Alhuseen Omar Alsayed[1], Muhammad Binsawad[2], Jawad Ali[3(✉)], Ahmad Shahrafidz Khalid[3], and Waqas Ahmed[4]

[1] Deanship of Scientific Research, King Abdulaziz University, Jeddah 21589, Saudi Arabia
aoalsayd@kau.edu.sa

[2] Faculty of Computer Information Systems, King Abdulaziz University, Jeddah 21589, Saudi Arabia
mbinsawad@kau.edu.sa

[3] Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia
jawad.ali@s.unikl.edu.my, ahmads@unikl.edu.my

[4] UniKL Business School, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia
waqas.ahmed@s.unikl.edu.my

Abstract. In Trusted Computing, the client platform is checked for its trustworthiness using Remote Attestation. Integrity Measurement Architecture (IMA) is a well-known technique of TCG based attestation. However, due to static nature of IMA, it cannot be aware of the runtime behavior of applications which leads to integrity problems. To overcome this problem several dynamic behavior-based attestation techniques have been proposed that can measure the run-time behavior of applications by capturing all system-calls produced by them. In this paper, we have proposed a system call based technique of intrusion detection for remote attestation in which macros are used for reporting. Macros are used to denote subsequences of system calls of variable length. The basic goal of this paper is to shorten the number of system calls by the concept of macros which ultimately reduces the processing time as well as network overhead.

Keywords: Remote attestation · Dynamic behavior · Intrusion detection · Trusted computing

1 Introduction

Nowadays the world is executed by computing technologies, security is exceedingly important. Existing IT and computing infrastructure become more intricate than the past. Many technologies have been developed, such as cloud computing, software as a service (SaaS), cloud formation, e-commerce, and virtualization, etc. These technologies facilitate the end users as well as the IT staff like

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 132–144, 2021.
https://doi.org/10.1007/978-3-030-55180-3_10

server maintainer and software developers. On the other side, the enhancement in these complex software stacks results in open doorways for new vulnerabilities. The question arises here is to ensure that the remote machine while communicating with, is trusted or not?

Trusted Computing is a known concept in today's computing infrastructure which helps in integrating hardware-based security in the existing security framework [1]. The most important feature of TCG technology is to embed hardware root of trust inside the computing platforms. For this purpose, TCG introduces a cryptographic co-processor chip called *Trusted Platform Module* (TPM) [2]. There are a number of tamper-proof locations called *Platform Configuration Registers* (PCRs) inside TPM. These PCRs can store platform configuration in the form of cryptographic hashes. SHA-1 is used as a one-way hash function that cannot be removed or changed by any software application. These stored hashes inside PCR can further be reported to a remote system in a secure channel. The process by which these hashes are reported to a remote platform in order to build trust is called *remote attestation*.

Remote Attestation is one of the known features of Trusted Computing which can be used to verify the integrity of remote systems, in order to build trust between them. There are several kinds of remote attestation techniques i.e. Static remote attestation [3] and dynamic behavior-based remote attestation [4–7]. A well-known technique for remote attestation is *Integrity Measurement Architecture*(IMA). Reiner Sailer et al. [3] designed an integrity measurement system and implemented it into Linux. IMA considers to be the first technique for remote attestation. In IMA the hashes of every executable loaded for execution are calculated and stored to a log file called SML (Stored Measurement Log). When the system boots, a chain of trust is formed by first computing the hash (SHA-1) of BIOS stored in TPM. The BIOS then passes the control to Boot Loader. The Boot loader measures and calculates hash of the Operating System kernel and stores the hash in PCR-10. The kernel is then responsible for loading further executables, such as init and libraries. The kernel first measures the hash of every executable and then allows the Operating System for loading. A log of these hashes gets maintained in a log file called stored measurement Log (SML). All these hashes are then aggregated and stored in TPM using PCR. After each calculated hash a PCR-extend function is called for concatenating the previous hash with the current hash to form a single hash. In order to verify the system remotely, the challenger first sends the attestation request. In response attesting system sends an attestation token which includes PCR-10 value and SML to challenging party. The challenger calculates all values of SML in the same order and compares with PCR-10, so if both the values are same then the remote party is considered to be trusted. But due to the static nature of IMA there are some limitations. It can only measure the load time measurements and cannot be well informed of runtime behavior. Another weakness of IMA is, it cannot have resistance to buffer overflow attacks and return-oriented programming [8,9]. Therefore, there is a need for dynamic behavior attestation mechanism to handle the said issues.

The most well-known technique that is used to measure the dynamic or runtime behavior of an application is through sequences of system calls generated by an application during its lifetime [9,10]. These techniques are effectively working on host-based security as intrusion detection systems (IDS), but there is no implementation in remote attestation scenarios where it is capable to report the behavior measurement to a remote party for verification. However, every single application generates a huge number of system calls in a small time-stamp [11]. This results in more processing time on the target platform that ultimately leads to a network overhead during transmission.

Outline: The rest of the paper is organized as follows: In Sect. 2 a description regarding our contribution to existing techniques of remote attestation is mentioned. Section 3 discussed the background and some of the literature studies about remote attestation techniques along with its limitations. In the same section we also provided some study of the intrusion detection systems to understand our proposed architecture. Further in Sect. 4 we define behavior in terms of macros and Sect. 5 elaborate Linux Security Module structure. Section 6 describes our proposed architecture of remote attestation along with analysis with the existing techniques. Finally Sect. 7 concludes the paper.

2 Contribution

In this research, we have proposed an existing technique of intrusion detection presented by [12] in a remote attestation framework in order to address the above problems. Afterward, we investigate how to reduce system call log by using the concept of macros. The log reduction will result in below:

- Increasing the efficiency of behavior measurement on remote end.
- Reducing log being sent to a challenger for verification would decrease the network overhead.

3 Background

3.1 Trusted Platform Module

Trusted Platform Module (TPM) has been introduced by TCG [13] specifications as a basic component. TPM is a cryptographic co-processor chip that secures the data and provides hardware root-of-trust. Almost all vendor's laptops and desktops have TPM inside of them. TPM has several components inside that are RSA Engine, Random Number Generator (RNG), SHA-1 & HMAC Engine, CPU, Volatile & Non-volatile memory RSA is a hardware engine being used in public-key cryptography and is one of the essential requirements for TPM. It is used to generate new keys for signing purposes as well as encryption and decryption of other keys. Additionally, TPM can support different key size that is 512 bits, 1024 bits, and 2048 bits. The SHA-1 algorithm is used to compute 160-bit hashes of data. RNG is used to create keys as well as nonces (numbers

used once) during attestation. It can also be applied through the use of software-based RNG which gets sourced from hardware RNG. HMAC is used as a keyed hash function that incorporates a cryptographic key for converting unkeyed hash function to keyed hash function. It is used for both to verify data integrity as well as authentication of a user. Authdata is a 160 bit secret code produced from the new-key which is generated inside of TPM. All these above functions are related to cryptographic capabilities of TPM.

There are a number of shielded locations inside TPM called *platform configuration registers* (PCRs). It plays a very important role in completing the process of remote attestation securely and reliably. PCRs are used to store measurements in the form of hashes. There can be a total number of 16 or 24 PCRs depending on the specification that is used to design TPM. PCRs 0 to 7 are reserved for pre-execution which shall build a chain of trust before the OS gets control. Several functions are used in TPM based operations. One of the important functions used while remote attestation is *PCR_Extend* that can append and aggregate the hashes in TPM.

Protection through hardware means that the information from the user will be stored accurately and signed by the TPM Key-pair normally called storage root key (SRK). This particular key is attached to TPM and could not get back from TPM. SRK can be used either directly to encrypt the data or for securing other keys called storage keys. SRK can be washed from BIOS through a TPM specific instruction i.e. TPM-ForceClear. The public endorsement key is used for binding data and can only be unbinded by private pair of public key. Similarly, sealing is done through public key as in binding but the unsealing operation is different from binding. A hint or some special instruction is given for unsealing e.g. nonce.

3.2 Dynamic Behavior Attestation

Integrity Measurement Architecture (IMA) [3] is a well-known technique of static remote attestation for verifying target platform by reporting the hashes of applications. Hashes of applications help out IMA to verify the trusted state of the target system upon the request of challenger. Various issues came out due to this hash-based technique, one of them is highly rigid target domains. As a solution to these problems variety of attestation techniques have been proposed. Such techniques are based upon the runtime attitude of applications, data structures, and system call sequences.

Jeager et al. [8] proposed an extension to the IMA called Policy Reduced Integrity Measurement Architecture (PRIMA). It tried to overcome the issues of IMA that is:

- It computes load time measurements of code which does not accurately reveal the runtime behaviors of an application.
- There is no support to verify the integrity of some specific application rather than the whole system that needs to be verified.

– IMA cannot reduce the list of measurement by measuring those applications which are associated with the target application.

PRIMA measures the integrity of target applications by information flow. It does not measure only the code but also has awareness of information flow between processes. Its prototype is implemented through SElinux policy for providing the information flow that would produce some nature of dynamism in attestation mechanisms. The additional requirements for information flow are load time MAC policy and trusted subjects, mapping between MAC policy and code. The author proved that it can attest CW-Lite (short version of Clark-Wilson). Moreover, PRIMA approach addresses some issues found in previous techniques such as false negative attestation, false positive attestation and also decrease the number of necessary measurements. However, there are some drawbacks in this technique as given:

– PRIMA is still dependent on hashes of code, policy, and files.
– It cannot capture the dynamic behavior of an application.

Gu et al. [9] proposed remote attestation on program execution which is a step towards dynamic behavior-based attestation. They capture the behavior of the application by collecting system calls with hardware root-of-trust. Although there are several solutions for monitoring behavior of application which provide security for the system, but the software-based security is not so feasible than hardware-based solution. However, there are some limitations to this approach.

– Enormous number of system calls can cause performance overhead on the target to be measured.
– Reporting of this immense number of system calls results in network overhead.
– Solely system calls cannot give any meaning in verifications, unless there is some sort of patterns i.e. sequence of system calls that capture the malicious behavior.

Loscoco et al. [14] developed a framework called Linux Kernel Integrity Measurement (LKIM) which is considered as a step towards behavioral attestation. It examines some critical running data structures and plots them on a graph for decision making. LKIM is based on contextual inspection that is used for measuring the components of running kernel. It can execute in both environments: as a user-process in base environment and domain in the hypervisor environment. LKIM monitors the running kernel so that it measures the components in the current state. Although this is quite a better approach but some limitations are still attached to it.

– As it measures the hashes of running kernel and produces a big amount of logs, in a result the network traffic increases and becomes a bottleneck over the network.
– LKIM analyzes kernel data structure at runtime that needs an extra amount of time for processing on the target end. However, due to the static nature of IMA it cannot be aware of the runtime behavior of applications which leads

to integrity problems. To overcome this problem several dynamic behavior-based attestation techniques have been proposed which can measure run-time behavior of applications by capturing all system-calls produced by them [9,10,15].

3.3 Intrusion Detection System

To monitor activities of a network or a computer system and to analyze them as whether the system or network is acting in malicious or normal way is known as Intrusion Detection System (IDS). Depending on classification there are two major types of intrusion detection system i.e. Host Based IDS & Network Based IDS [16]. Host based intrusion detection system (HIDS) provides protection and security against an individual host. Network intrusion detection system (NIDS) is used to monitor and protect traffic from the entire network and generate alarms or response against malicious attempt.

Based upon the detection techniques of intrusion detection system, there are two main categories of IDS i.e. Misuse IDS & Anomaly based IDS. Misuse IDS stores the known signature of intrusions. Whenever an action occurs and it matches with the previously experienced signature, is considered as misuse detection [17]. It reports the event which is matched as intrusion. Although this approach is consider to be better for accuracy with low false positive, but it cannot take action (detect) on new or unmatched pattern. In addition, while adding new signatures regularly will lead to performance overhead on the underlying system. To overcome this issue, anomaly based detection system is based on the behavior of system which creates normal behavior profiles for the system. When an event occurs in the system, it observes and gets compared with the normal behavior profile and report the [17] unusual deviations as intrusion. User behavior can be created or profiled through statistical methods, inductive pattern generation, data mining, machine learning and neural networks.

Kosoresow et al. [12] shows his preliminary work for analyzing system calls sequences for normal and abnormal behavior. They figured out *macros* which are generated by taking common prefixes, suffixes and repeating strings in a system call trace. Each macro consists of variable length pre-defined patterns of system calls and every application has its own set of macros.

Forrest et al. [18] introduces the idea of using short sequences of system calls of runtime (privileged) processes, which results in generating a stable signature for normal profile. Every program in execution produced a sequence of system calls and determined by their order in which they are executed. For any significant program, there will be a trace of system calls which have not been observed. However, short-range of system call sequences are notably consistent and results in defining normal behavior. To create a separate profile of normal behavior for each process, authors define system calls in the form of fixed length short sequences i.e. 5, 6 and 11.

4 Defining Macros as a Behavior

When an application runs, it generates system calls for performing different tasks. These system calls are matched against pre-defined macros to determine the sequence of macros. Macros are the regular pattern occurring frequently in a system call traces. In this case, its size is variable and each macro is denoted by an alphabet i.e. A, B, C, D to Z [12]. In this proposed research the traces of system calls are matched against the pre-defined macros which result in producing macros. For example, if an application generates a trace, it will be checked in the list of macros and finally generate a sequence of macros. Figure 1 illustrates the mechanism of producing macros from a system-calls sequence.

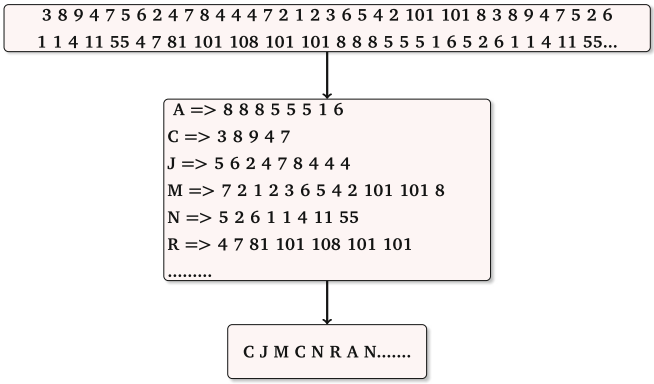


Fig. 1. System calls to macros transformation

5 Linux Security Module (LSM)

The most important feature provided by Linux kernel is the Linux Security Module (LSM). By default Linux is set up with *Discretionary Access control* (DAC) system. DAC is the first model towards access control mechanism and adopted by the Linux OS. DAC is an owner-centric security model which means that it restricts or grants access to an object by the policy defined by owner of the object [19]. For example, a user A creates a file then A will decide that what kind of access to file is provided to other users i.e. groups or others. A root user also called as privileged user has all the access rights on system. In case, if malicious user got login as root so there is no restriction in DAC to prevent it.

The LSM is a general framework based on Mandatory access control (MAC) which provides a base to third party security modules for implementing their own defined policies for carrying out any action in the system. The actions are determined by designers of the framework which required authorization. These actions are first passed through the LSM framework before the kernel finished

the task on behalf of an application. Since, in our proposed work we are trying to intercept calls with the help of a custom LSM module. This Module first creates macros from system calls trace and stores in a log file called *security-fs*. Before storing these macros we will take measurement in the form of hashes and store them to PCR. This process will be carried out through the attestation module in our proposed framework. Every application has its own fixed possible number of system calls that are generated during execution. We define macros of application on the target platform in a database. With the help of this macro database, we will identify the known and unknown macros generated from the trace.

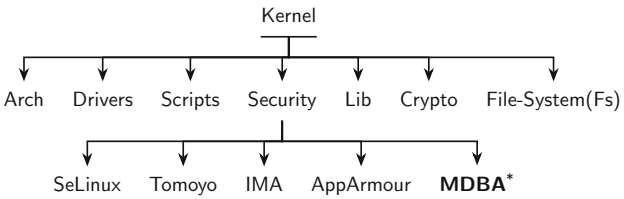


Fig. 2. Linux kernel directory structure

6 Proposed Architecture

The proposed architecture as shown in Fig. 3 consists of two entities i.e. target & Challenger. The first and focused entity in architecture is the target or client platform. The measurement process of application behavior is performed on a target platform which will further be used for verification on challenger-side. The second entity is the challenger who wants to know the trustworthiness of the client system. On target-end, a new custom LSM module named *Macro based Dynamic Behavior Attestation* (MDBA) is proposed as shown in Fig. 2, which will directly be connected with LSM hooks for creating macros rather than performing mapping between system-calls and LSM hooks. In contrast to earlier remote attestation mechanisms where every system call is considered to be measured, this proposed technique first creates macros that are pre-defined in the database and calculates the measurements (Hashes) of these macros. Further, these measurements will be stored in SML and their aggregated hash value will be stored in PCR.

6.1 Reporting Macros for Verification

Generally, a remote attestation mechanism establishes in request and response manner. To perform the attestation process, the first challenger will send an attestation request along with nonce to the target system for verifying trustworthiness. The target system receives an attestation request and prepares a

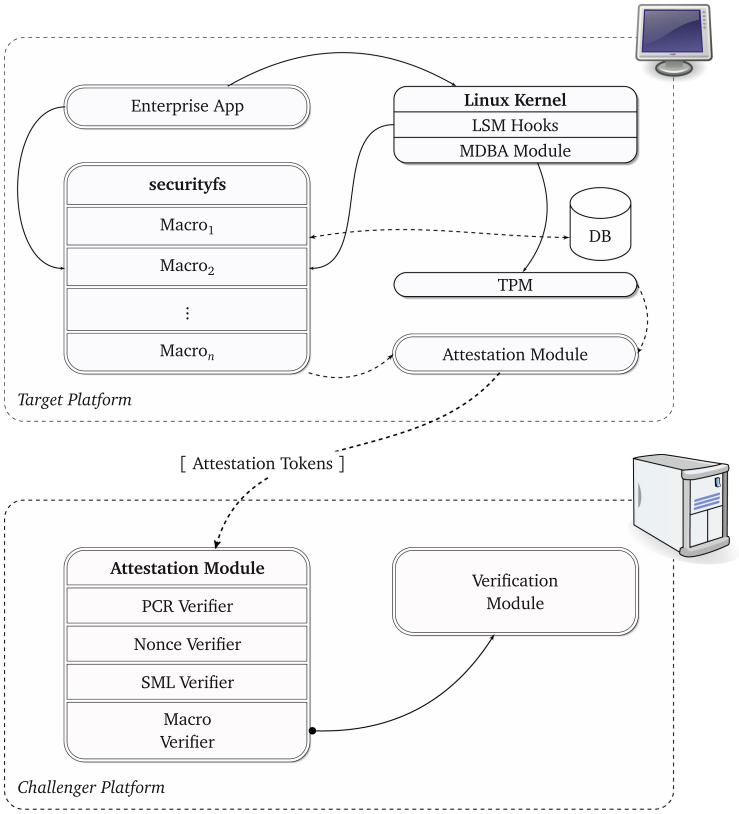
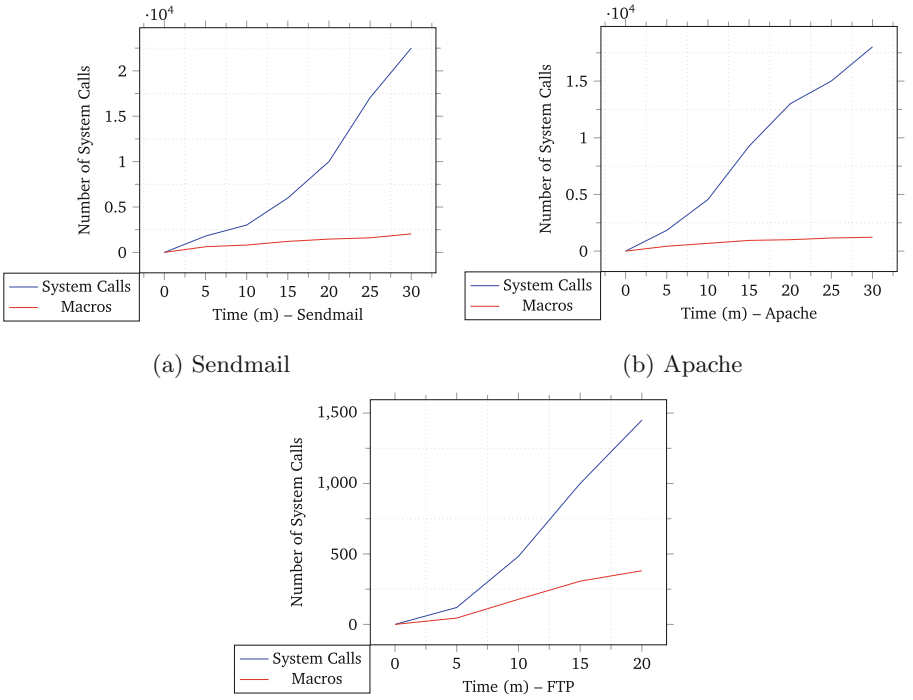


Fig. 3. Proposed architecture

response that contains all measurement logs along with the PCR-12 Quote. The TPM prepares this response at the target end. TPM first calculates the PCR-Composite structure over specifics PCRs. There are two PCRComposite structures calculated by attestation module: one is PCR-10 which is used to store system static measurement, while the other PCR-12 is used to store the measurement of macros of the client application. Afterward, TPM calculates hashes for each structure and appends these hashes to a fixed value along with the nonce received from the challenger side in attestation request. TPM then sign the values of PCR Composite using the private part of the Attestation Identity Key (AIK). Finally, the challenge-response is made and sent to the challenger that contains PCR Composite structures with their digital signatures. Now when the response is received, the challenger first recomputes hashes from the log (SML) and compares them with the PCR Quote, If both the values match to each other then the target will be considered trustworthy. This process will take place through the attestation module on the challenger end.

6.2 Comparisons of Results with Existing Techniques

Generally, every application generates a very large number of system calls in a short time. As a result, the system call log will also increase to an unbounded size. Reducing the log by introducing our proposed technique is one of the main objectives of this research. As discussed earlier, our technique maps the system calls traces of an application against the *macros* (variable length system call sub-sequences).



(a) Sendmail (b) Apache

(c) FTP

Fig. 4. System call VS macros comparison

Different applications have been analyzed to show the improvement over the existing system calls based techniques as shown in Fig. 4. Initially, a *sendmail* application is selected for testing as a target application (cf. Fig. 4a). After running the send-mail applications for 5 min, it generates almost 1800 system calls while their corresponding macros were 625 which is three times less than the actual number of system calls. This improvement will reduce the measurement size as well as a performance by lesser SHA-1 and PCR-Extend operations in TPM. After executing the application for a further 15 min, it generates about 6000 system calls while 1200 macros were seen.

And after half an hour the application produces 22500 system calls and the final value for macros was 2100. Further, *FTP-Client* application has been considered for testing. In the first five minutes of FTP session, it produces 120 system calls while 45 macros were found. After watching for 15 min it generates 1000 system calls while the number of macros was 300 as shown in Fig. 4c. Finally, *apache* application is monitored for about 30 min. Apache generates almost 18000 system calls while their corresponding macros were found almost 1250 as illustrated in Fig. 4b. The overall results from these applications show a significant decrease in the number of measurements. Furthermore, Table 1 shows the significance of our proposed approach in terms of minimizing the log. The reduction in system calls will also decrease the reporting log by sending only macros rather than the whole system calls log.

Table 1. System calls and macros log size comparisons

Times (min)	Target application	System call log size	Macros log size
5	FTP	0.31 KB	0.23 KB
15	FTP	14.34 KB	10.90 KB
25	FTP	24.77 KB	11.43 KB
5	SendMail	5.75 KB	4.20 KB
15	SendMail	19.50 KB	14.78 KB
25	SendMail	36.23 KB	21.63 KB
5	Apache	3.61 KB	0.41 KB
15	Apache	20.10 KB	9.63 KB
25	Apache	38.47 KB	20.03 KB

7 Conclusion

In this paper, we have presented a new technique of remote attestation which utilizes an existing intrusion detection system to measure the dynamic behavior of the remote application. We have discussed the workflow of the model along with the implementation plan in detail. We have studied the existing dynamic behavior attestation techniques and figure out their limitations which are the main bottleneck of these techniques to be implemented in the real scenarios. Using this simple technique proposed we can record and measure the dynamic nature of the remote platform. The most prominent aspect of our proposed solution is that system calls traces of an application are matched against a variable-length pattern of system calls called *macros*. Afterward, measurements of the macros are extended in TPM and their log is maintained in the SML-store measurement log. Representation of system calls by using macros reduced measurement and their log file sizes to an optimal size.

Our future work includes the real-time applications of the proposed architecture in various use-cases. We will give a detailed analysis of different other applications and show the usability of this work. We will provide an open-source implementation for getting feedback from the research community.

References

1. Mitchell, C., Mitchell, C., Mitchell, C.: Trusted Computing. Springer, Heidelberg (2005)
2. Bajikar, S.: Trusted platform module (TPM) based security on notebook PCS-white paper. Mobile Platforms Group, Intel Corporation, 20 June 2002
3. Sailer, R., Zhang, X., Jaeger, T., Van Doorn, L.: Design and implementation of a TCG-based integrity measurement architecture (2004)
4. Ali, T., Ali, J., Ali, T., Nauman, M., Musa, S.: Efficient, scalable and privacy preserving application attestation in a multi stakeholder scenario. In: International Conference on Computational Science and its Applications, pp. 407–421. Springer (2016)
5. Syed, T.A., Jan, S., Musa, S., Ali, J.: Providing efficient, scalable and privacy preserved verification mechanism in remote attestation. In: 2016 International Conference on Information and Communication Technology (ICICTM), pp. 236–245. IEEE (2016)
6. Ali, T., Zuhairi, M., Ali, J., Musa, S., Nauman, M.: A complete behavioral measurement and reporting: optimized for mobile devices. In: COMPSE 2016 - 1st EAI International Conference on Computer Science and Engineering (2017)
7. Syed, T.A., Ismail, R., Musa, S., Nauman, M., Khan, S.: A sense of others: behavioral attestation of unix processes on remote platforms. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ICUIMC 2012, pp. 51:1–51:7. ACM, New York (2012). https://doi.org/10.1145/2184751.2184814
8. Jaeger, T., Sailer, R., Shankar, U.: Prima: policy-reduced integrity measurement architecture. In: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, pp. 19–28. ACM (2006)
9. Gu, L., Ding, X., Deng, R.H., Xie, B., Mei, H.: Remote attestation on program execution. In: Proceedings of the 3rd ACM Workshop on Scalable Trusted Computing, STC 2008, pp. 11–20. ACM, New York (2008). https://doi.org/10.1145/1456455.1456458
10. Liang, G., Ding, X., Deng, R.H., Xie, B., Mei, H.: Remote attestation on function execution (2009)
11. Nauman, M., Ali, T., Rauf, A.: Using trusted computing for privacy preserving keystroke-based authentication in smartphones. Telecommun. Syst. **52**(4), 2149–2161 (2013)
12. Kosoresow, A.P., Hofmeyr, S.A.: Intrusion detection via system call traces. IEEE Softw. **14**(5), 35–42 (1997)
13. Trusting Computing Group (2014). http://www.trustedcomputinggroup.org/. Accessed 19 July 2014
14. Loscocco, P.A., Wilson, P.W., Pendergrass, J.A., McDonell, C.D.: Linux kernel integrity measurement using contextual inspection. In: Proceedings of the 2007 ACM Workshop on Scalable Trusted Computing, STC 2007, pp. 21–29. ACM, New York (2007). https://doi.org/10.1145/1314354.1314362
15. Kil, C., Sezer, E.C., Azab, A.M., Ning, P., Zhang, X.: Remote attestation todynamic system properties: towards providing complete system integrity evidence. In: IEEE/IFIPInternational Conference on Dependable Systems & Networks, DSN 2009, pp. 115–124. IEEE (2009)
16. Axelsson, S.: Intrusion detection systems: a survey and taxonomy. Technical report (2000)

17. Debar, H., Dacier, M., Wespi, A.: Towards a taxonomy of intrusion-detection systems. Comput. Netw. **31**(8), 805–822 (1999)
18. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of selffor unix processes. In: Proceedings of 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE (1996)
19. Benantar, M.: Access Control Systems: Security, Identity Management and Trust Models. Springer, Heidelberg (2006)

The Effect of Using Artificial Intelligence on Performance of Appraisal System: A Case Study for University of Jeddah Staff in Saudi Arabia

Ahmed Alrashedi[1(✉)] and Maysam Abbod[2]

[1] College of Business, University of Jeddah, Jeddah, Saudi Arabia
aalrashde@uj.edu.sa
[2] Department of Electronic and Computer Engineering, Brunel University London, Uxbridge, UK
maysam.abbod@brunel.ac.uk
https://www.brunel.ac.uk/people/Maysam-abbod

Abstract. Despite the interest of developed countries in relying on artificial intelligence in the performance of their work, developing countries are at the beginning of this path. The researcher has noted that the use of modern technologies, especially artificial intelligence, did not give the necessary attention in government business and many private sector companies, so it was necessary to draw attention to the importance of using it to develop all businesses. This study focuses on clarifying the importance of using artificial intelligence (AI) technology in the process of evaluating the employee's performance to increase the effectiveness of performance appraisal for all organisational levels to make strategic decisions which affect the objectives of the organisation. This study has used a questionnaire that looked at participants' views and attitudes towards technology and its usefulness in the appraisal process. Knowing the opinions of employees and their supervisors will enable to demonstrate the need to use AI to raise the efficiency of the performance appraisal system and develop its processes for the employee's benefits and the general interest at the organisation. Most of the answers were between agree and strongly agree, and on the other side, there was a very small percentage that did not believe the importance of using artificial intelligence in works, and this may be due to their feeling of resisting change and the desire to work a way they used to. This paper begins by providing demographic details and background information on the participants, followed by an examination of the reliability of the extracted factors/metrics. The extracted factors will then be described individually using general descriptive statistics (such as repetition percentages) to look at their respective elements and see the level of the agreement created by the participant. Following this descriptive part of the analysis, this paper seeks to conduct an in-depth analysis of the results using various inferential statistics to test group differences using demographic and background details of the participants (e.g. age, gender, education). This study investigates the impact of the use of AI on the performance appraisal model, and to examine the effect of AI on the technology adoption.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 145–154, 2021.
https://doi.org/10.1007/978-3-030-55180-3_11

Keywords: Artificial intelligence · Performance appraisal · Software benefit · Technology adoption

1 Introduction

Performance appraisal is an essential task for any manager to determine the performance of his subordinates and their achievement of business objectives in [1]. It imposes physical and psychological effort on managers having to investigate the factors that influence the employee performance in [2]. They found that there are five factors with positive impact on performance appraisal system, namely: implementation process, interpersonal relationships, rate accuracy, informational factors, and employee attitudes. in addition, there are two evaluation process' approaches, one is technical regarding the accuracy and comprehensiveness of employee-related information from multiple locations within the organization, and the other is social regarding employee acceptance of evaluation results in [3]. Technically, an organisation can use artificial intelligence (AI) to collect data from multiple sites within the organisation on how an employee performs his work. By linking these available data, line managers could come up with appropriate and objective proposals that predict individual performance and help the organisation to make negative or positive decisions regarding employee's behaviours. Furthermore, on the social side, employees will be more convinced and confident with the AI system due to its accuracy and objective in comparison to traditional assessments in [4].

Performance appraisal is essential for both employees and organisation. The employee would know the exact level of their performance because self-assessment can be inaccurate. On the other hand, the organisation would be aware of the performance level of its staff to decide who requires training to improve and develop his performance or learn new skills for his work. Furthermore, evaluation results are essential for determining who is being penalised, rewarded or transferred to another job that fits with his abilities. Both sides can be beneficiaries. In [5]. has argued that in order for organisations to achieve competitive advantages, they must keep abreast of society technological developments that enable them to develop their business, so it is necessary to utilize technologies such as AI to develop and improve the performance appraisal system and encourage employees by providing accurate and honest results of the evaluation system in [6].

The reliance on AI technology will help to obtain a vast amount of information related to the employee by logging previous evaluations and views of the former and current managers, which gives an integrated employees picture; therefore, the assessment will be written in an accurate and objective way. The results of the evaluation should be evaluated, analysed and reviewed for all staff to categorise available competencies based on business needs for all departments in the organisation in [7].

This study investigates the impact of the use of AI on the performance of the appraisal model, and to examine the effect of technology adoption on the use of artificial intelligence. See Fig. 1.

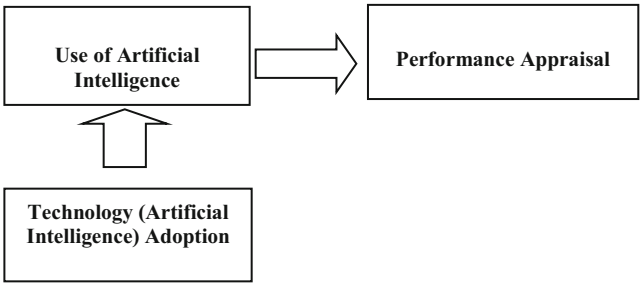


Fig. 1. An image showing technology adoption that affects the use of AI which affect performance appraisal.

2 Literature Review

The performance appraisal system is one of the most critical systems to any organization to know the level of employees' performance and their ability to achieve its goals with the highest efficiency to be able to face its competitors either in the local or global market. In [5]. has argued that In order to achieve competitive advantages, organizations must keep abreast of technological developments in society that enable them to develop their businesses and implementers, so it was necessary to take advantage of the technology of artificial intelligence to develop and improve the performance evaluation system and to encourage employees to feel the accuracy and fairness of the evaluation system in [6].

2.1 Artificial Intelligence

AI is a computer algorithm has the capability to learn in [1]. It is a system that simulates complex problem solving based on its capabilities of symbolic thinking, flexibility and explanation in [8]. In [9], have defined AI as a scientific field designed to simulate the behaviour of the human brain by devices, both of them are considered as an information processing machines. In [9, 10], have proposed a framework of intelligent human resources information based on the use of hybrid AI tools such as automated learning and knowledge-based approaches so, that data can be collected without human intervention, stored, accurately summarized, processed to derive new information to support the decision-making process. Dependence on AI will accomplish business more quickly, correctly and cost less, leading to the competitive advantage of the organization. It will also assist human resources management in planning, reporting, assessing policies, forecasting needs, training and forecasting staff performance, all those without human intervention. It is possible to say that AI is a new generation of technology that will make the perfect machine capable of simulating human behaviour entirely and whose components are small in [11].

It is known that AI is not about one technology; it is a set of different techniques to obtain high-quality products and services in less time and cost. Human resources professionals can take advantage of these technologies for improving the system such as recruitment, selection, training, development, performance appraisal, compensation and reward in [12]. In [13], have discussed how computers contribute to business performance and economic growth. It is not vital to use technology like others, but it is crucial to take advantage of it in [14].

In [15], have believed that Artificial intelligence will soon be able to do administrative tasks that consume more than half of managers time in jobs, coordination control and evolution employees will be faster and better and at a lower cost. They have also convinced that managers who deal with AI such as researcher, explorer, analyst, evaluator, in addition to giving them different scenarios to solve problems. All these will make employees trust his evolution. By using smart technology, the system will receive a tremendous amount of information about the staff, which can be viewed in one place with accuracy and speed that is unexpected at the least time and cost. It is much better than traditional methods. In [16], have found that the system of mutual memory helps to share knowledge among bosses and between employees and their bosses which positively would affect organizational performance.

2.2 Performance Appraisal

In [2], have viewed factors influencing employee performance appraisal system. They found that there are five factors that have positive or negative impact on performance appraisal system, which are implementation process, interpersonal relationships, rate accuracy, informational factors, and employee attitudes. In [17], have found that because employees are the most critical resource, they must be developed and improved performance to achieve the organizational goals, and therefore, human resource management through managers should measure staff performance annually and frequently to ensure their in [18].

Because of the lack of evaluator's managerial skills, this will negatively affect the outcome of the evaluation and lose its impact on employees, the organization has to rely on AI for performance appraisal. In [19], has argued that organizations can design performance appraisal systems per the legal regulation of organizational framework that makes it necessary to use an accurate, comprehensive, transparent and objective scientific method. These features will only be available in AI. This will be able to make the best decisions about the employees' performance. These decisions can be trusted and easy for employees to accept and managers to discuss their findings objectively with staff. These decisions will be based on all information related to team past and current performances which make them objective and acceptable to all employees. They also can reduce doubts about their accuracy and relevance and will be better than if it relies solely on a traditional approach that depends only on supervisor's ability to remember all bad or good situations to write employee evaluation in [3].

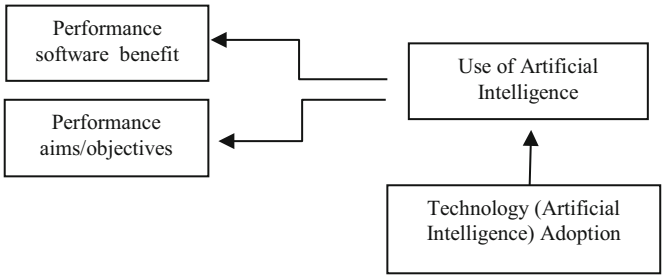


Fig. 2. Research hypotheses and research model.

3 Research Hypotheses and Research Model

This study tests the following hypotheses (see Fig. 2):

H1: Technology (AI) Adoption will positively affect the Use of AI.
H2: Use of AI will positively affect Performance software benefit of performance appraisal.
H3: Use of AI will positively affect Performance aims/objectives of performance appraisal.

4 Procedures and Data Collection

A total of 339 employees were randomly selected from universities and companies in Saudi Arabia, namely King Abdulaziz University, King Khalid University, Taif University, Umm Al-Qura University, Saudi Aramco, Saudi Arabian Airline. There were 113 females and 226 males. The aggregate number of participants was aged between 18 to 55 years old.

The questionnaire was constructed to cover three distinct dimensions, namely Performance Appraisal (11 items), Utilisation of Artificial intelligence (11 items) and Technology Adoption factors (8 items). The aspects were constructed in a way to answer the research questions/aims asset declared in the introduction. Employee responses were obtained using a 4-point Likert-type scale where one = strongly disagree, and 4 = strongly agree.

5 Data Collection

5.1 Participants' Demographic Data

This section provides a general description of the background and demographic characteristics of the sample used in this study. Overall, 339 took part, 66.6% were males, and 33.3% were females. Their education and qualification varied where slight majority of the participants had PhD level of skill (34.3%) followed by bachelor's degree (30.2%),

and 27.3% stated they have a master's degree. Only 2.9% had a diploma level of qualification, and 5.3% explained they had completed high school level of education. As of the organisation they work for the majority of 66.9% work in the government sector, and 33.1% worked in the private sector. Finally, participants' nationalities varied were the vast majority of 83% were Saudis, and 17% were non-Saudis. Egyptians (10%) and Tunisians (3.8%) were the two main nationalities represented other than Saudis. Table 1 shows the frequencies and the percentages of all variables.

5.2 Reliability

Cronbach's Alpha test was conducted to measure the consistency in answers across items within each of the dimensions (internal reliability) and each of the factors produced following Factor Analysis. This test enables the research to judge how reliable each of the scales is, i.e. how consistently items measure for the same thing. Cronbach's alpha is a coefficient that ranges between 0 and 1 in size, where value close to 0.70 or above are considered acceptable. As can be observed from Table 2, all dimensions can be regarded as reliable.

6 Results and Analysis

6.1 The Impact Technology Adoption Factors on the Use of AI

Linear regression was conducted to examine the effects of technology adoption factors on use of AI. Table 3 shows that a significant relationship exists between the dependent variable Use of Artificial intelligence and the independent variables Technology (AI) Adoption, where $F = 276.853$ and P-value < 0.001 (the coefficient of determination being 45.0%), but that high significant relationship exists thereof with technology adoption factors and the use of AI.

6.2 Impact of Use of Artificial Intelligence on Performance Appraisal

Two factors were analysed, namely performance software benefits, and performance aim/objectives. For factor 1, linear regression was conducted to examine the impact of using AIon performance software benefit. The results presented in Table 4 show that a significant relationship exists between the dependent variable performance software benefit and the independent variables use of Artificial intelligence, where $F = 90.075$ and P-value < 0.001 (the coefficient of determination being 21.0) which proves the Hypothesis H2 is actual "Use of AI will positively affect Performance software benefit of Performance Appraisal".

For factor 2, linear regression was conducted to examine the impact of the use of AIon performance aims/objectives. Table 5 shows that a significant relationship exists between the dependent variable Performance aims/objectives of the determination being 40.9%), which supports that the hypothesis H3 is valid "Use of Artificial intelligence will positively affect Performance aims/objectives of Performance Appraisal".

Table 1. General demographic and background details of the participants.

Age	n	%	Gender	n	%	Nationality	n	%
18–25	10	2.9	Male	226	66.3	Saudi	283	83.0
26–30	22	6.5	Female	113	33.1	Egyptian	34	10.0
31–35	50	14.7	Missing	2	0.6	Jordanian	4	1.2
36–40	66	19.4	Education	n	%	Tunisian	13	3.8
41–45	45	13.2	High School	18	5.3	Sudanese	3	0.9
46–50	60	17.6	Diploma	10	2.9	Palestinian	1	0.3
>50	88	25.8	Bachelor	103	30.2	Lebanese	1	0.3
			Master	93	27.3	Missing	2	0.6
			PhD	117	34.3	Grouped Nationality	n	%
Organisation	n	%				Saudi	283	83.0
Government	226	66.3				Non-Saudi	56	17
Private	113	33.1				Total	339	100

Table 2. Cronbach's alpha as a measure for internal reliability.

Dimension/scale	Items	Cronbach's alpha
Performance appraisal	11	0.815
Performance and software benefit	6	0.793
Performance aims/objectives	5	0.755
The utilisation of artificial intelligence	11	0.910
Technology adoption factors	8	0.903

Table 3. Linear Regressing between dependent variable "use of technology" and independent variable "AI technology adoption" in the study.

	Unstandardized coefficients		Standardized coefficients	t	Sig.	ANOVA		R^2
	B	Std. Error	Beta			F	P-value	
Constant	1.656	0.165		10.009	0.000	276.853	<0.001	0.450
Technology (AI) adoption factors	0.626	0.038	0.670	16.639	0.000			
a. **Dependent variable:** use of technology								

Table 4. Linear regressing between dependent variable "performance software benefit" and independent "use of AI" variable in the study.

	Unstandardized coefficients		Standardized coefficients	t	Sig.	ANOVA		R^2
	B	Std. Error	Beta			F	P-value	
Constant	2.681	0.192		13.937	0.000	90.075	<0.001	0.210
Use of AI	0.413	0.044	0.458	9.491	0.000			
Dependent variable: performance software benefit (Factor 1)								

Table 5. Linear regression between Dependent Variable (Performance aims/objectives) and Independent (Use of AI) variables in the study.

	Unstandardized coefficients		Standardized coefficients	t	Sig.	ANOVA		R^2
	B	Std. Error	Beta			F	P-value	
Constant	0.950	0.215		4.421	0.000	234.214	<0.001	0.409
Use of AI	0.745	0.049	0.639	15.304	0.000			
Dependent variable: performance aims/objectives (Factor 2)								

Table 6. Summary of hypothesis testing.

Hypothesis	Specification	Results
H1	AI technology adoption positively affect the use of AI	Supported ($\beta = 0.626$, $p < 0.01$)
H2	Use of AI positively affect performance software benefit of performance appraisal	Supported ($\beta = 0.413$, $p < 0.01$)
H3	Use of AI positively affect performance aims/objectives of performance appraisal	Supported ($\beta = 0.745$, $p < 0.01$)

7 Conclusion and Discussion

This study to investigate the impact of Use of AI on Performance software benefit and Performance aims/objectives, also to examine the effects of AI technology adoption on use of AI. In summary, as shown in Table 6, the results of the linear regression analyses confirmed the three hypotheses. Use of AI has high significant effect on Performance software benefit and on Performance aims/objectives also Technology (AI) Adoption had the most definite impact on the Use of AI. This study has shown that, Technological Artificial Intelligence Role in Raising the Efficiency of Performance Appraisal System.

Based on the results of this study which have showed positive impact of AI on the performance appraisal program, the researcher is currently working on designing a program that helps human resources management to evaluate employees' performance in order to take advantage of artificial intelligence in obtaining an integrated image of employee to include all the relevant information and data since their entry to company until the present, with the ability to predict future career that make benefit to both company and employees.

References

1. Kornienko, A.A., Kornienko, A.V., Fofanov, O.B., Chubik, M.P.: Knowledge in artificial intelligence systems: searching the strategies for application. Proc.-Soc. Behav. Sci. **166**, 589–594 (2015)
2. Antunes, P., Herskovic, V., Ochoa, S.F., Pino, J.A.: Structuring dimensions for collaborative systems evaluation. ACM Comput.Surv. (CSUR) **44**(2), 8 (2012)
3. Arrays, J.I.: Electronic performance measurement systems: feasibility of dynamic performance measurement systems: a case study. Doctoral dissertation, Nottingham Trent University (2017)
4. Jewels, T., Ford, M.: The development of a taxonomy of desired personal qualities for IT project team members and its use in an educational setting. J. Inf. Technol. Educ.: Res. **5**(1), 285–298 (2006)
5. Orlikowski, W.J.: Using technology and constituting structures: a practise lens for studying technology in organizations. Organ. Sci. **11**(4), 404–428 (2000)
6. Sholihin, M.: How does procedural fairness affect performance evaluation system satisfaction? (Evidence from a UK Police Force). Gadjah Mada Int. J. Bus. **15**(3), 231 (2013)
7. Daoanis, L.E.: Performance appraisal system: it's implication to employee performance. Int. J. Econ. Manag. Sci. **2**(3), 55–62 (2012)
8. Metaxiotis, K., Karagiannis, A., Askounis, D., Psarras, J.: Artificial intelligence in short term electric load forecasting: a state-of-the-art survey for the researcher. Energy Convers. Manag. **44**(9), 1525–1534 (2003)
9. Rodríguez, D., Hermosillo, J., Lara, B.: Meaning in artificial agents: the symbol grounding problem revisited. Mind. Mach. **22**(1), 25–34 (2012)
10. Masum, A.K., Beh, L.S., Azad, A.K., Hoque, K.: Intelligent human resource information system (i-HRIS): a holistic decision support framework for HR excellence. Int. Arab J. Inf. Technol. **15**(1), 121–130 (2018)
11. Agrawal, A., Gans, J.S., Goldfarb, A.: Artificial intelligence: the ambiguous labour market impact of automating prediction. J. Econ. Perspect. **33**(2), 31–50 (2019)
12. Jain, V.K., Kumar, S., Fernandes, S.L.: Extraction of emotions from multilingual text using intelligent text processing and computational linguistics. J. Comput. Sci. **21**, 316–326 (2017)
13. Brynjolfsson, E., Hitt, L.M.: Beyond computation: Information technology, organizational transformation and business performance. J. Econ. Perspect. **14**(4), 23–48 (2000)
14. D'Amico, E., Leone, C., Hayrettin, T., Patti, F.: Can we define a rehabilitation strategy for cognitive impairment in progressive multiple sclerosis? A critical appraisal. Multiple Sclerosis J. **22**(5), 581–589 (2016)
15. Kolbjørnsrud, V., Amico, R., Thomas, R.J.: How artificial intelligence will redefine management. Harv. Bus. Rev. **2**, 1–6 (2016)

16. Choi, S.Y., Lee, H., Yoo, Y.: The impact of information technology and transactive memory systems on knowledge sharing, application, and team performance: a field study. MIS Q. 855–870 (2010)
17. Iqbal, N., Ahmad, N., Haider, Z., Batool, Y., Ul-ain, Q.: Impact of performance appraisal on employee's performance involving the moderating role of motivation. Oman Chap. Arab. J. Bus. Manag. Rev. **34**(981), 1–20 (2013)
18. Klimoski, R.J., London, M.: Role of the rater in performance appraisal. J. Appl. Psychol. **59**(4), 445 (1974)
19. Jawahar, I.M.: Correlates of satisfaction with performance appraisal feedback. J. Labor Res. **27**(2), 213–236 (2006)

Traffic Accidents Analysis with the GPS/Arc/GIS Telecommunication System

Arbnor Pajaziti(✉) and Orlat Tafilaj

Faculty of Mechanical Engineering, University of Pristina, Pristina, Kosovo
arbnor.pajaziti@uni-pr.edu, orlattafilaj@gmail.com

Abstract. The purpose of this paper is to describe the application of the GPS/Arc/GIS telecommunication System in transport and traffic to identify the accident blackspot locations in the city of Pristina, Kosovo. It is also outlined the basics of GPS (Global Positioning System)–GIS (Geographic Information System) technology and the application of this technology in the research area. Further, the analysis are made of the application of GPS-GIS technology in traffic and transport with possibilities for future use of this technology for the municipality of Pristina. In this paper, the maps have been made with GPS data and have been compared with the conventional way of accident records. Here is shown how GPS/Arc/GIS combination gives the accurate black spot identification, rather than relying on assumed data for the location.

Keywords: Telecommunication System · GIS · GPS · Traffic accident

1 Introduction

Nowadays, there is a rapid development of science and technology. In this paper about Remote Sensing technology, GPS technology, and GIS technology has been discussed. These technologies have been developed with fast steps, but what makes them more special is the integration with each other. Another practical case of this integration is the use of GPS data in a GIS. Therefore in the full sense of the word, one can say that RS, GIS and, GPS technologies have complemented each other and that their development would not be possible without each other (Fig. 1). The integration of these technologies can be conceived in the following models:

- Linear model,
- Interactive model,
- Hierarchical model, and
- Complex model.

In this case, GPS data can be exported directly to a GIS database to update it or build new databases [1]. This data can be a point, linear, or even superficial. Their geometric properties must be transformed into those stored data in the GIS database before integration.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 155–163, 2021.
https://doi.org/10.1007/978-3-030-55180-3_12

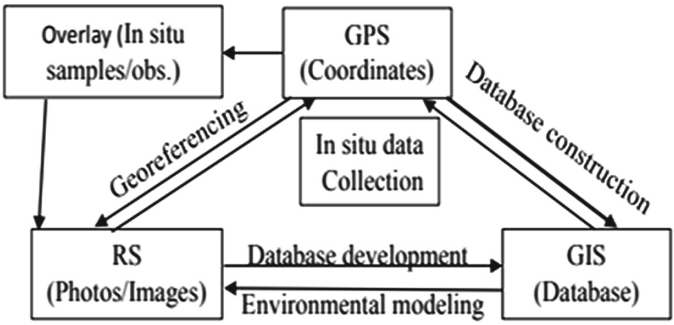


Fig. 1. Complex integration model [4].

1.1 Existing Applications

The value of GIS, GPS and RS integration, and terrain lies in those applications that require comprehensive and georeferenced data up to seconds or instantly. These applications include resource management and environmental monitoring, emergency response, mobile mapping, logistics, research, monitoring, conservation applications, technology transfer, education, etc.

Other capabilities, such as improved flood forecasting and global mobile communications while facilitating, are almost within reach. Decisions related to floodplain management require a digital high-resolution elevation model (DEM) and floodplain GIS data layers. A high-resolution DEM can be generated using digitally scanned photos in conjunction with highly accurate ground control from a GPS survey [2].

The mapping of linear features (egg roads, pipelines, power lines, river networks, coastlines, etc.) [3] and to some extent the characteristics of the area, is only achievable with GPS marking data along with the features of their perimeter. Aerial photographs can be used to locate trees accurately and to create suitable maps that highlight individual trees and other landmarks [5]. Highway, and Railway Maintenance, Softcopy Photogrammetry and Utility Mapping [6].

2 GPS-GIS Application Analysis in Traffic

For the study area in this case the city of Pristina a schematic diagram which contains spatial, numerical, and textual data has been presented. All these data are then integrated into the GIS (Fig. 2). The diagram shows the importance of other layers for a quality operation such as:

- Topographic service and land use,
- Transport network infrastructure,
- Socio-Economic and Demographic study,
- Traffic study,
- All of the information on traders and pollution.

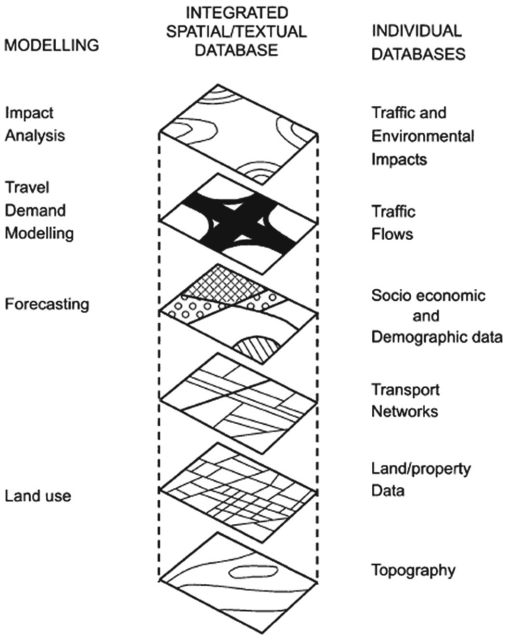


Fig. 2. Concept of overlapping data layers in GIS [7].

What is important at this point is also the use of the field of modeling and traffic planning [8].

The advantages of using GPS and GIS include:

- Position data and speed data from GIS,
- Practical data analysis as well as the possibility of using different types of data,
- GPS devices are very simple and easy to install in all vehicles.

3 Application of GIS to Traffic Accidents

GIS has found great use in the treatment and analysis of traffic accidents. The application of GIS in this field is based on data collection and then their analysis. The application of this technology is mostly in the identification of locations with the most common problems, and with the largest number of traffic accidents. With this technology, it is possible to present the location of accidents on the map and to present more information about the accident. What makes this technology more special is that even a simple user can get information about the accident. Below some steps that enable the identification and treatment of accidents have been presented.

– Step 1: Collection and compilation of accident data
  The first step in producing a national database is to compile an accident form in the relevant police districts.

– Step 2: Computerize and process accident records
  The structure of data, processing and data analysis is schematically summarized
– Step 3: Identify and prioritize accident blackspot locations
  There are a number of location identification systems that can be used for referencing and locating accidents. The site identification system can be based on either road maps being digitized and converted to DCM format or MAAP textual output.
– Step 4: Prioritize and diagnose black spots
  Black point rankings can be performed as Accident map ranking, node analysis, accident point rank, accident cost ranking, and kilometer-cost analysis.
– Step 5: Detailed diagnosis and countermeasures
  This step describes the in-depth and well-developed methods of diagnosis of selected black spot countermeasures. The field study is conducted to capture the near misses, vehicle speeds and pedestrian flows and their maneuvres.
– Step 6: Evaluation Techniques
  There are also a number of evaluation techniques currently used in evaluating security interventions. The selection of each technique depends on the nature of the work, the availability of data, and the accuracy required. Among the techniques available are cumulative plot techniques, and multivariate analysis.

4 Methodology

Through in Fig. 3, a summary of the methodology in five phases has been presented.

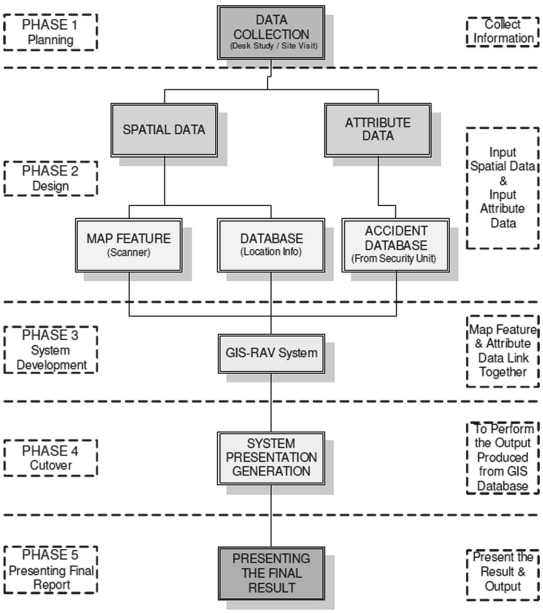


Fig. 3. Study overview [9].

The first phase is about planning, the second phase is about designing, the third is about developing the system, the fourth is about presenting the system, and the fifth is about presenting the final report.

5 Accident Analysis in the City of Pristina

The city of Pristina reports the highest number of accidents in the country, but the highest number of fatalities has been recorded in non-urban areas. The contributing factors that contribute to the increase in accidents are mainly the failure to adapt to the signs provided by the Law on Road Traffic, which mainly affects the speed of accidents with fatal consequences, failure to maintain distance and failure to adapt to driving conditions of the road, or climatic conditions.

5.1 Accident Statistics for 2016, 2017 and 2018

Comparison of the road accident statistics by type for 2016, 2017, and 2018.

The analysis shows that the highest number of accidents occurred during 2016 compared to the other two years, the following are the data collected for the three years presented in Fig. 4.

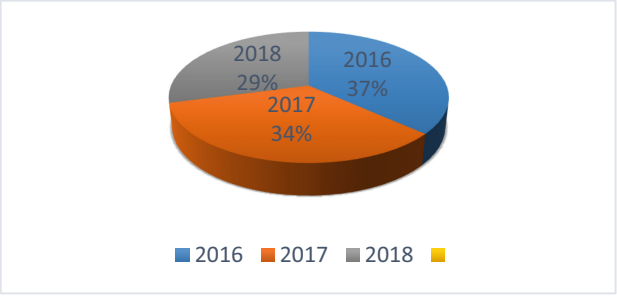


Fig. 4. Total accident status for 2016, 2017, and 2018.

In 2016 there were 8160 in Kosovo, in 2017 there are 7604 in Kosovo, and in 2018 there are 6494.

5.2 Accident Statistics with Injured and Dead for 2016, 2017 and 2018

During 2016, 4393 persons were injured in a road accident in Kosovo, while in 2017 they received 4604 injuries, while in 2018 they received 4467 injuries. In the comparison for deaths, 2018 is the year in which most people died in total 43, compared to 2017 where 35 died, and in 2016 37 died which is shown in Fig. 5. The classification of the accidents by location is presented in Fig. 6.

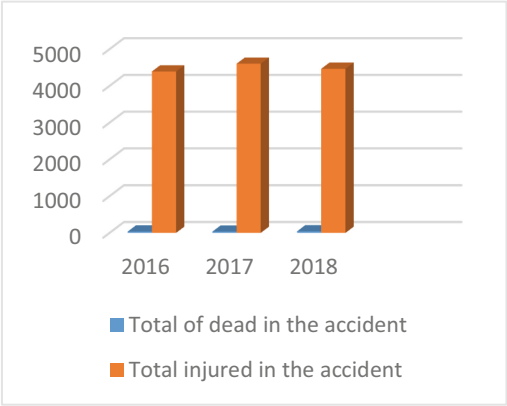


Fig. 5. Classification of dead and injured accidents for 2016, 2017, and 2018.

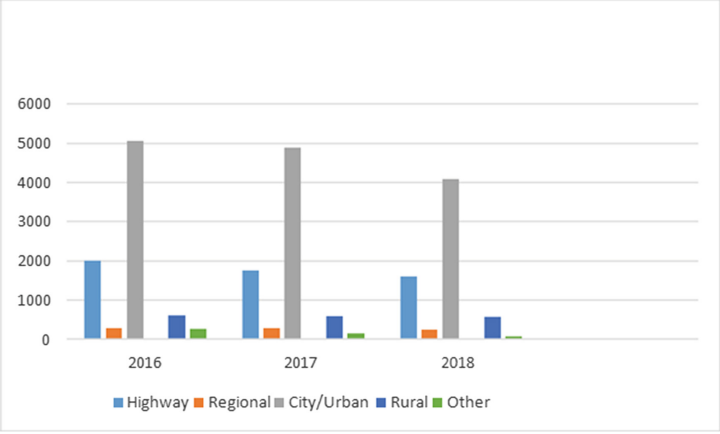


Fig. 6. Classification of accidents by location.

6 Results

After collecting accident data, a report can be made to ArcGIS, namely ArcMap where the accident sites could be identified precisely where we could have a realistic representation of the highest accident locations. The following is an analysis of how this data can be imported into ArcGIS software. As a practical case, five accidents were received in the municipality of Pristina to present the procedure of implementation of this process.

The following figures show the data on injuries and heavy injuries resulting from traffic accidents referring to Table 1. The processing of all accident data would be a great help for analysis in the Municipality of Pristina. These accident data are presented by mapping and road description (Fig. 7) and by satellite view (Fig. 8).

Table 1. Accident data.

Description	Longitude	Latitude	Date
Accident with body injury	42,685,511	21.15899	11.05.2019
Accident with body injury	42,661,364	21.16491	23.08.2019
Accident with heavy body injury	42,669,838	21.17051	24.01.2019
Accident with body injury	42,675,194	21.17328	25.02.2019
Accident with heavy body injury	42,653,899	21.15916	23.11.2019

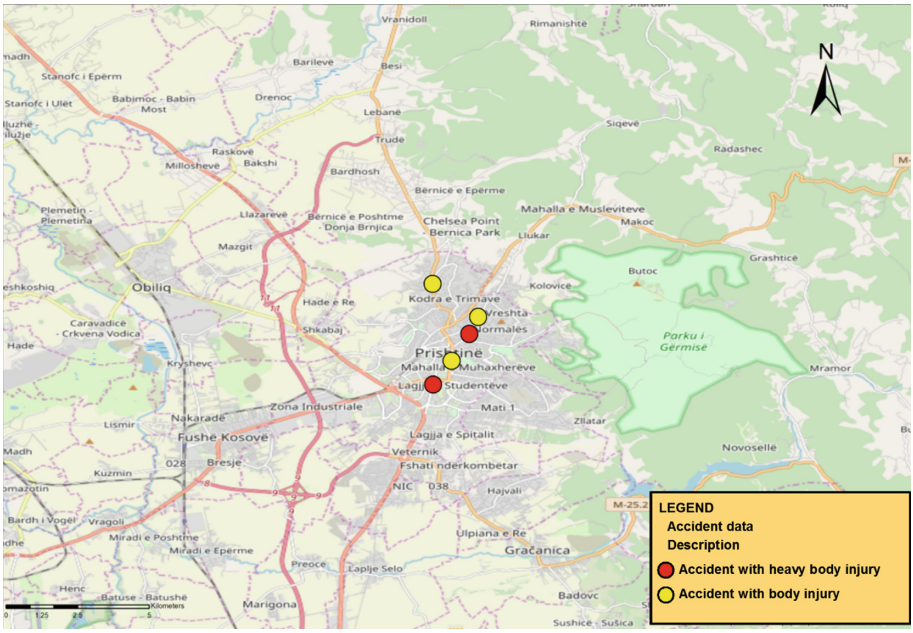


Fig. 7. Presentation of accident locations and its description [10].

The number of crashes is different every month, of course. We hope that by identifying where and when crashes occur throughout Pristina, we might be able to help prevent some of them.

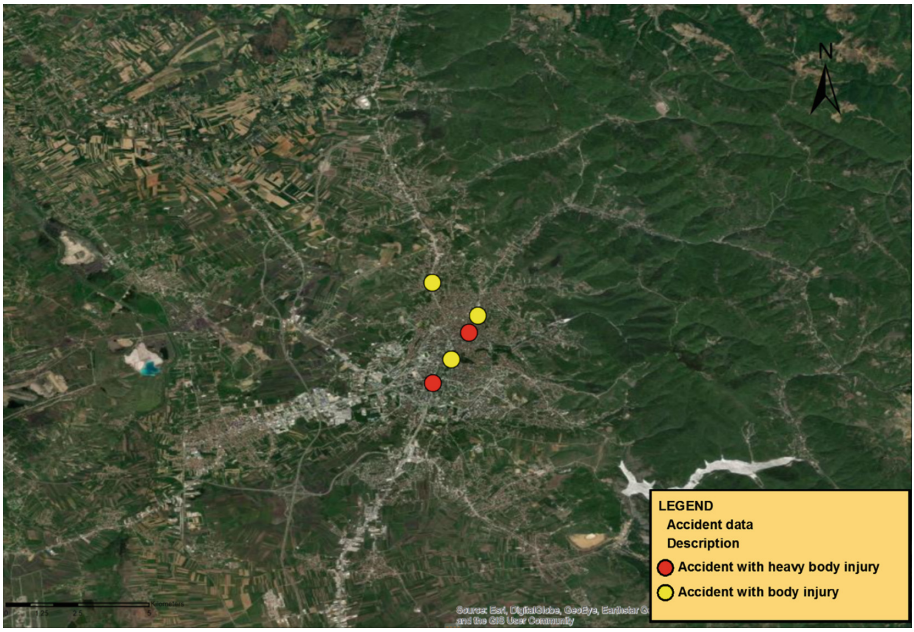


Fig. 8. Map of accident locations and its description on the satellite map [10].

7 Conclusion

This paper described the integration of GIS and GPS technology in particular in the treatment of traffic accidents. This technology is being used in all countries of the world to treat accidents and facilitate the presentation of the analysis of these data through a map medium. Nowadays, it is used in all sectors of information and communication. This integration of this technology enables us to receive data on time, store all information in GIS, modeling and analysis of this data, as well as the presentation of this data and for the public.

A GIS-based application was chosen as the best option to improve accuracy and timeliness in accident location priorities. Its initial advantages are its user-friendly software, the ability to quickly and accurately locate locations on a map.

8 Future Work

This system offers the possibility of covering traffic and other data with affordable prices for all cities and countries that use this technology.

This technology is constantly being improved, especially with the development of information technology, trying to establish and integrate the most advanced technologies for simulation models.

The Web GIS system is also expanding day by day with its potential to enable the public to receive quality and fast information promptly on time.

References

1. Bor, W.T.: The use of GPS in GIs applications. J. Geogr. Sci. **17**, 77–85 (1994)
2. Sugumaran, R., Davis, C., Meyer, J., Prato, T., Fulcher, C.: Web-based decision support tool for floodplain management using high-resolution DEM. Photogramm. Eng. Remote Sens. **66**(10), 1261–1265 (2000)
3. Cooper, R.D., Mccarthy, T., Raper, J.: Airborne videography and GPS. Earth Obs. Mag. **4**(11), 53–55 (1995)
4. Gao, J.: Integration of GPS with remote sensing and GIS: reality and prospect. Photogramm. Eng. Remote Sens. **68**(5), 447–454 (2002)
5. Kane, B., Rayan III, H.D.P.: Locating trees using a geographic information system and the glopal positioning system. J. Arboric. **24**(3), 135–143 (1998)
6. Novak, K.: Data collection for multi-media GIS using mobile mapping systems. Geod. Info Mag. **7**(10), 30–32 (1993)
7. Waters, N.: Transportation GIS: GIS-T. In: Longley, P., Goodchild, M., Maguire, D., Rhind, D. (eds.) Geographical Information Systems: Principles, Techniques, Applications, and Management, pp. 827–844. Wiley, New York (1999)
8. Hysenaj, M.: Geographical Information Systems. Shkodër, Albania (2011)
9. Liang, L.Y., Ma'soem, D.M., Hua, L.T.: Traffic accident application using geographic information system. J. East Asia Soc. Transp. Stud. **6**, 3574–3589 (2005)
10. Tafilaj, O.: Application of GPS–GIS in traffic and transportation for the municipality of Pristina. University of Pristina (2019)

The Hybrid Design for Artificial Intelligence Systems

R. V. Dushkin[1](✉) and M. G. Andronov[2](✉)

[1] VoiceLink LLC., Milashenkova st. 4a, Moscow 127322, Russian Federation
roman.dushkin@gmail.com
[2] Lomonosov Moscow State University, Leninskiye Gory st. 1, Moscow 119991, Russian Federation
mihandronov@gmail.com

Abstract. The article discusses approaches to intelligent systems building based on a hybrid paradigm implemented by combining the bottom-up (neural network) and top-down (symbolic) approaches to the design and development of artificial intelligence systems. The scheme of the hybrid intelligence system device is described, its architecture, the purpose and functionality of its components, the principles of operation and the use of its subsystems are explained.

Keywords: Artificial intelligence · Hybrid AI · Intelligence system

1 Introduction

Artificial intelligence is a multidisciplinary field of research, the rapid development of which began in the XX century. In the mid-century, there were attempts to create artificial general intelligence. However, it quickly became apparent that intelligence development *in silico* is a genuinely massive challenge. Later through the joint efforts of mathematicians and physiologists an artificial neuron model and the first artificial neural network were made. To the disappointment of scientists, no mind emerged in them. It became clear that the presence of a neural network did not guarantee its intelligence, and the mind was most likely due to some obscure synergistic effects within a network of many millions of neurons [1].

There are two approaches in artificial intelligence building. The first one is known as top-down AI.

Examples of a top-down approach are technologies such as expert systems, decision support systems, knowledge bases, and inference engines [3]. The principles of top-down AI are expressed in the Newell-Simon hypothesis: "A physical symbol system has the necessary and sufficient means for general intelligent action" [2]. The direction of symbolic calculations based on the logic of syntactic manipulation of symbols is the most developed in this approach.

The second approach, defined by Marvin Lee Minsky, is called the "bottom-up AI". It is based on the assumption that it is possible to model natural low-level processes

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 164–170, 2021.
https://doi.org/10.1007/978-3-030-55180-3_13

occurring in the living brain. This area combines technologies such as artificial neural networks, evolutionary algorithms, and biocomputing [4].

The two paradigms described above are at the heart of any approach to the development of AI systems.

Both top-down AI and bottom-up AI approaches have their advantages and disadvantages. The benefits of the first one include the ease of interpreting and explaining the results obtained by the logical inference engine based on symbolic calculations. At the same time, it is not very easy to work with large amounts of data in a top-down approach. That is because of the need to prepare too large knowledge bases for the inference engine, which will also have to be built with linguistic variables whose value is not strictly defined [5]. Moreover, the question of training artificial intelligence systems based on a top-down paradigm is still rather open-a final formalism for describing the function of independent learning of intelligent systems has not been proposed [1].

The bottom-up approach has rather opposite problems: an artificial neural network can easily be trained on huge amounts of data, but it will be almost impossible to interpret the result. There is no complete understanding of how a trained artificial neural network (for arbitrary architecture) produces a particular output, so the network is a "black box" [6]. Therefore, it is difficult to verify the correctness of the results generated by the neural network, for example, in the case of incorrect data falling into the sample on which the network was trained. Also, artificial neural networks are fundamentally different from the neural network in the human brain: neurogenesis and retraining are continually going on in the human brain, together with other unclear effects that lead to the emergence of consciousness [7]. Therefore, artificial neural networks alone cannot claim intelligence in the human sense.

2 The Architecture of a Hybrid Intelligent System

The solution to the problem of constructing artificial intelligence systems may lie in combining the capabilities of the top-down and bottom-up approaches. One can try to build the architecture of a hybrid intellectual system while continuing to be inspired by the design of the human mind but on a more general level. The following figure graphically shows the general diagram of the interaction of components in a hybrid intelligent system based on such principles.

Sensors (affectors) transmit information to a neural network that converts data from the sensors to symbolic form, based on which a universal inference engine generates input data for a motor neural network. The result of the motor neural network is low-level commands that control the executive devices of the hybrid intellectual system. In this case, the sensors also perceive information about the state of the components of the system itself.

One can improve the circuit in Fig. 1, breaking the control system into two subsystems, and then the general scheme of the hybrid intelligent system will look like the one shown in Fig. 2:

Now the control system is split into two parts:

Reactive Management Subsystem. This system implements the usual control scheme. Signals from the sensors are processed in the control system, generating control actions

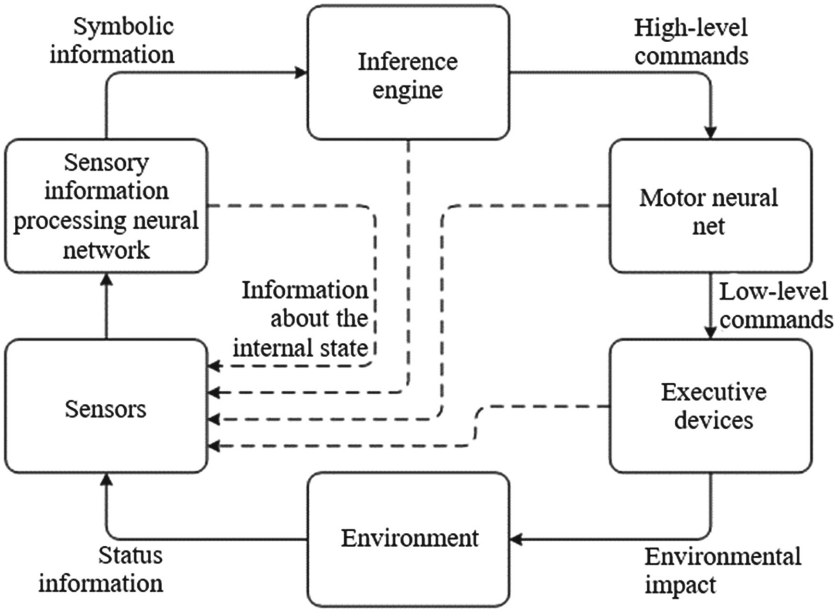


Fig. 1. General hybrid intellectual system architecture.

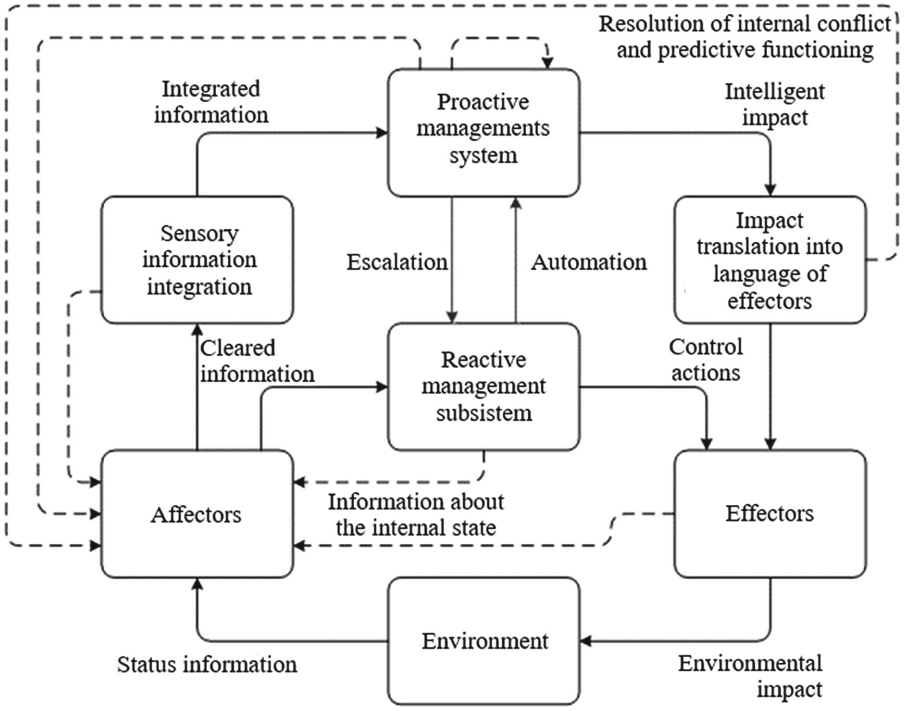


Fig. 2. Extended hybrid intellectual system architecture.

on the control object (the environment), that transmit through executive devices. This system is similar to the "reflex contour" in humans.

Proactive Management Subsystem. Adds layer, which serves to intellectualise the circuit of the automatic control system. This level allows the system to learn, predict its condition and environmental conditions, build planned actions, and also adapt to changing environmental conditions. This subsystem implements a hybrid system of "conditioned reflexes".

A management focus interchanges between the two systems. When the conditions of the system behaviour change, the proactive circuit creates a new pattern of behaviour and the control focus shifts to the reactive subsystem. Constant environmental conditions do not require the operation of the dynamic system. Therefore, a learned reaction is preferable. Such a situation is like the formation of a "conditioned reflex". On the other hand, when the environment or control object conditions change during the operation of the reactive subsystem, the control focus escalates to proactive to adapt and develop new patterns of system behaviour.

The functioning cycle of the hybrid system consists in sequentially performing the following steps:

Collection of input information from all sensors monitoring the parameters of the environment of the system. Different types of sensors, in this case, are separate modalities of a hybrid intelligent system perception.
Cleaning the collected information from noise and choosing a path for further processing. If the input information corresponds to any automatic patterns of system behaviour, then the control focus shifts to the reactive subsystem, which selects and executes a specific pattern.

The functioning cycle of the hybrid system consists in sequentially performing the following steps:

Integration of all modalities of the system perception into a single unit of the description of the environment if there is no automatic reaction. This block outputs a holistic picture of the perception of the environment that goes to the proactive control subsystem.
Formation of a new rule for the reactive system. This rule is a control action in a symbolic form, which is recorded in the reactive system and sent for execution. The proactive system derives the rule based on models of the system, its behaviour and environment using the mechanism of symbolic inference, so a person can easily interpret it.
Translation of the symbolic control action into a low-level language and transmittance to actuators that interact with the environment and the control object. Translation can be carried out by various mechanisms. For example, a neural network. After the execution of the command by the executive devices, the operation cycle ends.

Besides, there should be an implementation of control communications of all the hybrid intelligent system internal elements to its sensors. This creates adaptation mechanisms based on the constancy of the internal state of the system.

Finally, the connection from the proactive management subsystem to itself embodies the so-called "internal conflict", when the intelligent system can model various options for the development of a situation that affects the system itself. The cycle of evaluating and choosing an acceptable alternative runs until this internal conflict resolves.

3 Application Examples

An automatic control system evolves into a hybrid artificial intelligence system by intellectualisation, which means the increase of adaptability and autonomy [8]. Therefore, many well-known, but intellectualised in the hybrid paradigm control systems can serve as examples of the use of hybrid AI systems. The following examples are real examples taken from the authors' working practice.

Intellectual System of Traffic Control. Automated traffic control systems are designed to ensure safety and optimal road speed. With the massive introduction of uncrewed vehicles, intelligent transport systems become a logical continuation of the idea of automated traffic. Shifting the control mode from the traditional planning to adaptive traffic control throughout the entire street-road network of the city ensures the transition to intelligent transport systems. In this case, peripheral equipment objects and road traffic management equipment located on roads and roadside infrastructure facilities become the effectors of the intelligent control system. The system makes decisions on control actions depending on the parameters of traffic flows and forecasts given by the transport model, taking into account the development of the meteorological and road conditions [9].

Building Management Systems. Building management systems can be designed in the hybrid paradigm. A building may contain a large number of different sensors that monitor the state of its internal environment. Some particular environment parameters must remain constant. The presence of a reactive subsystem in the control system allows one to respond to known situations, but one reactive system will not cope with new unaccounted problems. Adding a proactive subsystem to the control system allows the latter to learn in the process of functioning and correctly respond to new changes that occur in the environment [8].

Intellectualization of Technological Processes. Almost any automated technological process can be implemented as a hybrid intelligent system. Like a building management system, a process control system needs a proactive control subsystem containing models of the control object and its operating environment. The proactive subsystem will increase the intellectuality of the process by predicting and planning control actions with learning in an automated mode based on a comparison of the forecast, plan and fact [10, 11].

Intellectualization of Online Learning. The process of online learning is based on the independent study by the student of the course materials and communication with other students and teachers. An analysis of the questions asked at the forums shows that most of the issues related to the topic of the course are typical. To intellectualise the online learning system in this case, one can use an intelligent agent (for example, a

chatbot) to automatically generate answers to students' questions in a natural language. Issuance of answers can be conducted in a personalised mode with automated training of an intelligent agent to answer questions to which the agent has not previously answered [12].

4 Conclusion

The architecture of the hybrid intellectual system presented in this work describes a separate class of intelligent systems, which is based on mimicking of individual aspects of the functioning of human intelligence. One can assume that the refinement of the proposed architecture with specific technologies in the implementation of such intelligent systems will increase the efficiency of the application of solutions based on artificial intelligence.

The examples of the application of the described architecture show that the proposed architecture and approach are universal enough to be applied in various problem areas. At the same time, these questions are still open, and additional studies on the opening opportunities, methods of application, and emerging effects are required.

References

1. Yashchenko, V.A.: Teoriya iskusstvennogo intellekta (osnovnye polozheniya) [Theory of artificial intelligence (key points)]. Matematicheskie mashiny i sisitemy [Math. Mach. Syst.] **1**(4), 3–19 (2011). (in Russian)
2. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall, Englewood Cliffs (1995)
3. Chernenko, V.V., Piskorskaya, S.Y.: Ekspertnie Sistemi. [Expert Systems.] Aktualnie problemi aviacii i kosmonavtiki. [Act. Problems Aviat. Cosmonautics] (8), 322–323 (2012). (in Russian)
4. Dushkin, R.V.: Obzor podhodov I metodov iskusstvennogo intellekta. [Overview of approaches and methods of artificial intelligence.] Radioelektronnie tehnologii. [Radioelectron. Technol.] (3), 85–89 (2018). (in Russian)
5. Nariniyani, A.S.: Nedoopredellennost v Sistemah Predstavleniya I Obrabotki Znaniy. [Under determination in knowledge representation and processing systems.] Isvestiya AN SSSR. Tehn. Kibernetika. [News of the Academy of Sciences of the USSR. Techn. Cybern.] (5), 3–28 (1986). (in Russian)
6. Tong, A., van Dijk, D., Stanley, J.S., Amodio, M., Wolf, G., Krishnaswamy, S.: Graph spectral regularization for neural network interpretability. In: ICLR, New Orleans (2019)
7. Steiner, E., Tata, M., Frisén, J.: A fresh look at adult neurogenesis. Nat. Med. **25**, 542–543 (2019)
8. Dushkin, R.V.: Osobennosti funkcional'nogo podhoda v upravlenii vnutrennej sredoj intellektual'nyh zdanij. [Features of the functional approach in managing the internal environment of intelligent buildings.] Prikladnaya informatika. [Appl. Inform.] **13**(6), 20–31 (2018). (in Russian)

9. Andreeva, E.A., Belkova, E.V., Dushkin, R.V., Zharkov, A.D., Kurochkin, E.A., Levin, N.V., Morozov, V.P.: Tematicheskij obzor Associacii Transportnyh Inzhenerov: sistemy adaptivnogo upravleniya dorozhnym dvizheniem i dorozhnye kontrollery. [Thematic review of the Association of Transport Engineers: adaptive traffic management systems and road controllers.] Izdatel'sko-poligraficheskaya kompaniya «KOSTA», Saint Petersburg (2017). (in Russian)
10. Ickovich, E.L.: Metody racional'noj avtomatizacii proizvodstva. [Methods of rational automation of production.] Infra-Inzheneriya Publ., Moscow (2009). (in Russian)
11. Dushkin, R.V., Koptev, A.P.: Avtomatizaciya delovyh processov pri pomoshchi Edinogo kompleksa avtomatizirovannyh sistem upravleniya predpriyatiem. [Automation of business processes with the help of a single set of automated enterprise management systems.] In: Collection of Abstracts of the International Scientific and Practical Conference 2008, INTEHMET, Saint Petersburg, pp. 33–34 (2008). (in Russian)
12. Dushkin, R.V.: Razvitie metodov adaptivnogo obucheniya pri pomoshchi ispol'zovaniya intellektual'nyh agentov. [The development of adaptive learning methods using intelligent agents.] Iskusstvennyj intellekt i prinyatie reshenij. [Artif. Intell. Decis. Making] (1), 87–96 (2019). (in Russian)

An Automated Approach for Sustainability Evaluation Based on Environmental, Social and Governance Factors

Ventsislav Nikolov(✉)

Technical University of Varna, Studentska 1, 9010 Varna, Bulgaria
v.nikolov@tu-varna.bg

Abstract. This paper shortly describes a new approach for evaluation of Environmental, Social and Governance factors and combines them into an overall ESG rating. The proposed approach is based on automated calculations, implemented in a software system, called Sustainability Evaluator, that provides ESG ratings for small and medium-sized enterprises and organizations by using of known ESG ratings of other companies.

Keywords: Sustainability · Environment · Social · Governance · Multifactor · Validation · Neural network

1 Challenge and Solution

The assessment of Environmental, Social and Governance (ESG) indicators plays an important role in investment analysis, affecting the reputation and trustworthiness of the financial market participants. Long-term sustainable development strategies require such analysis and for that reason ESG ratings are introduced. They are especially valuable when it comes to well-known companies and corporations. These ratings direct the attention toward socially responsible investments and facilitate the sustainable risk analysis.

In the past, several sustainability management standards, metrics and indices have been introduced, such as: Global Reporting Initiative (GRI), Sustainability Accounting Standards Board (SASB), ISO 26000 Social Responsibility, Dow Jones Sustainability Indices (DJSI), Responsible Business Alliance (RBA), etc. The significance of such standards has increased over time and they have been adopted by corporations and investors [2, 3].

ESG indicators measure the sustainable development of companies in different economy sectors and reflect whether corporate decisions and activities have taken into account multiple aspects concerning environmental protection, organizational effectiveness, social benefits, and so on (Fig. 1). The ESG rating generates a long-term estimation of a company's trustworthiness. The companies without ESG rating might not be very attractive to the potential partners and investors, that normally prefer to take as informed as possible decisions.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 171–182, 2021.
https://doi.org/10.1007/978-3-030-55180-3_14

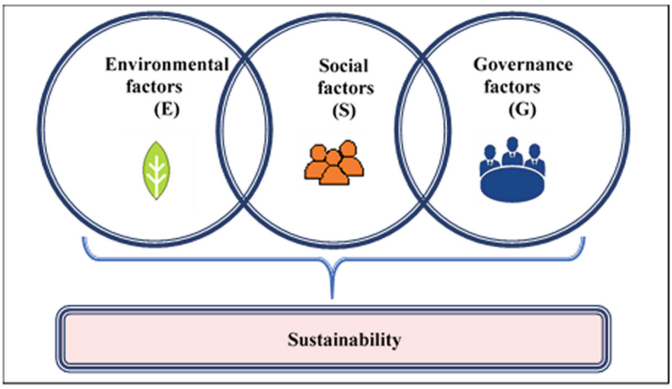


Fig. 1. Environmental, Social and Governance as indicators for sustainability.

The described methodology enables an automatic evaluation of ESG rating of states, organizations, companies, regions and municipalities, including small and less known companies. By analyzing a company's formal relations to other market participants, the methodology provides clear information on the target company and hence impacts investors' effectiveness, their decisions and strategies.

2 Innovation

The current state of the art is that ESG ratings are evaluated for certain companies only, by using the common approach, that is based on expert created questionnaires. The questionnaires are composed of hard and soft facts that have to be identified first and then assessed for each of the three ESG factors (Environmental, Social and Governance), after which a numerical score is determined for each factor. Finally, the numerical scores are combined into a single score – the ESG rating. Factors that are taken into account, as well as their weights, often depend on the business sector. While hard facts represent directly measurable and indisputable data, soft facts are based on different opinions, which can be subjective. For that reason, this approach is considered subjective as well.

The challenges of the ESG rating estimation mainly concern the process of automation. Thus, both the ESG ratings become objective and the automation allows the ratings to be generated in short time periods. The automation is based on multifactor analysis, which consists of considering the historical company market data (e.g. share prices, products prices, etc.) and building a mathematical model by analyzing the dependency from such data of other companies. Since the historical data are based on decisions of multiple market participants, this method can be considered objective, thus truly reflecting the current state of a company. The mathematical model can be used by each market participant that is interested in evaluating a target company (e.g. a partner, a customer, etc.).

3 The Multifactor Approach: Methodology

Suppose we are given a finite number of discrete time series, called factors or variables, with equal length. They can represent arbitrary physical, social, financial or other processes or indicators. As such, their values correspond to measurements with certain frequency for all factors. One series is considered as a target factor and the others are explanatory factors. Explanatory factors represent independent variables, while the target factor is a dependent variable (Fig. 2). The goal is to create a formula by which a series can be generated, using explanatory factors for a given historical period, which should be as close as possible to the given target series, using a chosen criterion [7, 8]. For simplification purposes, such a criterion can be the Euclidean distance between the given and generated target factor for all historical data points.

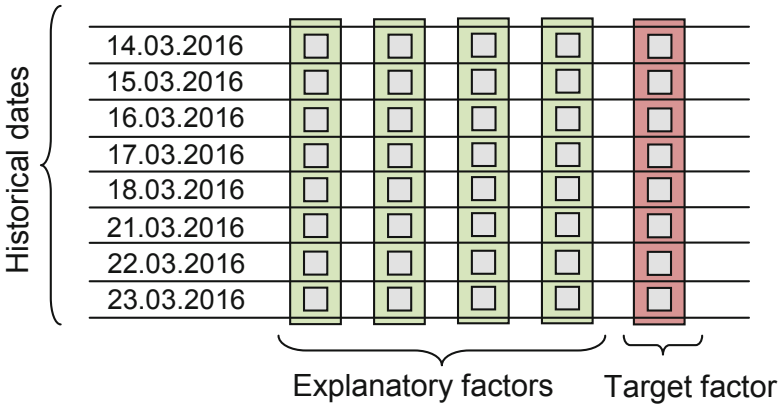


Fig. 2. Explanatory and target factor in multifactor modelling

Such a formula can be used for different purposes:

- Modelling of financial instruments. For example, an unknown composition of a stock index. Its model should be known in order to perform different calculations, such as simulations and Value at Risk (VaR) estimation.
- Sensitivity analysis, which evaluates the influence of changes of the explanatory factors on the target factor. This analysis can be quantitatively performed for one or more explanatory factors and the effect can be analyzed and practically interpreted.

According to our research, currently there is no software applications performing these operations in their clear form for the ESG rating evaluation. The formula can be in different forms, but in order to simplify the solution, the following polynomial form is used in our system:

$$y = \beta_1 f_1(f_1) + \beta_2 f_2(f_2) + \ldots + \beta_m f_m(f_m) + \beta_{m+1} \quad (1)$$

where $f_1, f_2, \ldots f_m$ represent arbitrary functions, called basis functions, $\beta_1, \beta_2, \ldots \beta_m$ represent numerical coefficients, called regression coefficients, and β_{m+1} is a free numerical term without an explanatory factor.

Formula Generation

The modelling stage starts with the selection of a target factor. In the practical multifactor modelling, all possible explanatory time series can participate in the formula. Since normally there are too many series that can be explanatory factors, a methodology for their selection must be chosen [9]. In our solution, a few alternative approaches are used, such as the selection of factors that most correlate to the target factor or minimally correlate to each other, etc.

When both target and explanatory factors are selected, the automatic modelling stage is performed by repeating the following steps:

1) Applying basis functions to explanatory factors;
2) Calculating the regression coefficients.

Performing the first step, in fact, produces new values for the explanatory factors, after which a new solution must be found by performing the next step. All this should be repeated as many times as needed in order to find the best combination of functions for the selected explanatory factors. In our software system, the first approach, does that randomly. The second implemented approach uses a more systematic procedure to find the best combination of functions. Considering that all functions can be applied to each of the selected factors, k^m combinations exist, where k is the number of basis functions and m is the number of explanatory factors. Usually, the explanatory factors are a few hundred and the basis functions are a few dozen. This means that, if the best combination must be found by brute force searching, there would be too many solutions to generate and calculate. That is why a heuristic approach is applied in our solution, using an evolutionary algorithm.

Finding the Best Combination of the Basis Functions

In practical solutions it is important for every experimental result to be reproducible. For that reason, our solution creates a main random generator that works with or without a seed value.

Initial Set of Candidate Solutions

Provided that the functions are positioned in a fixed order, the main aim of the algorithm is to find a sequence of basis function indices that are as good as possible in respect to the Euclidean error between the generated and given target factor. For this purpose, a random integer sequence generator has been created that generates the initial population of integer sequences. Applying the functions to factors and calculating the regression coefficients produces a set of target factors, which are then compared to the real target. Thus, in terms of the evolutionary algorithm, an individual, that is a candidate solution, is represented as a sequence of integers with a length that are equals to the number of explanatory factors, while the goodness of fit is the distance between the generated and a

given target factor. If a free term is being used, it is associated to a mock factor composed by values 1.0 for all historical dates.

Selection
Given a set of generated N individuals, the best L of them are selected. There are two alternative approaches: roulette wheel and truncation selection [10, 11]. The first is preferred in our solution as it allows each individual to continue the process regardless of the fact that its goodness of fit function produces poor result. Such individuals will just have a lesser chance to continue being part of the algorithm, even though it is not impossible.

Recombination and Mutation
Recombination is performed by splitting the selected L individuals in one point and combining the split parts randomly. In our implementation, the splitting point is randomly generated at every step within the interval from 25% to 75% of the individual's length, rounded to the nearest integer.

Coefficients Determination
The second step of the formula is calculating the regression coefficients, which is performed for every function combination to the explanatory factors. In our case, an ordinary least squares error is used, according to which coefficients are obtained in a matrix form, calculating the following matrix equation [9, 12–14]:

$$B = (A^T A)^{-1} A^T Y \tag{2}$$

where B is the matrix of the regression coefficients, A is the matrix of factors with applied basis functions and Y is the target factor.

After the coefficients are calculated, they are being used for the computation of the generated target factor:

$$\hat{Y} = A \times B \tag{3}$$

The distance between the generated and available target is:

$$d = \left\| Y - \hat{Y} \right\| \tag{4}$$

This distance can be calculated with or without a decay factor [15].

Coefficients Reduction
The formula terms with small coefficients can be removed, as they do not significantly influence the results. Removing small coefficients is optional in our solution and if it is performed the regression coefficients are calculated again.

Calibration
Using the generated formula for future calculations and modeling must be calibrated periodically and the formula must be reevaluated. This is needed, as with the progress of time, the accuracy of the formula decreases. The calibration process is shown in Fig. 3.

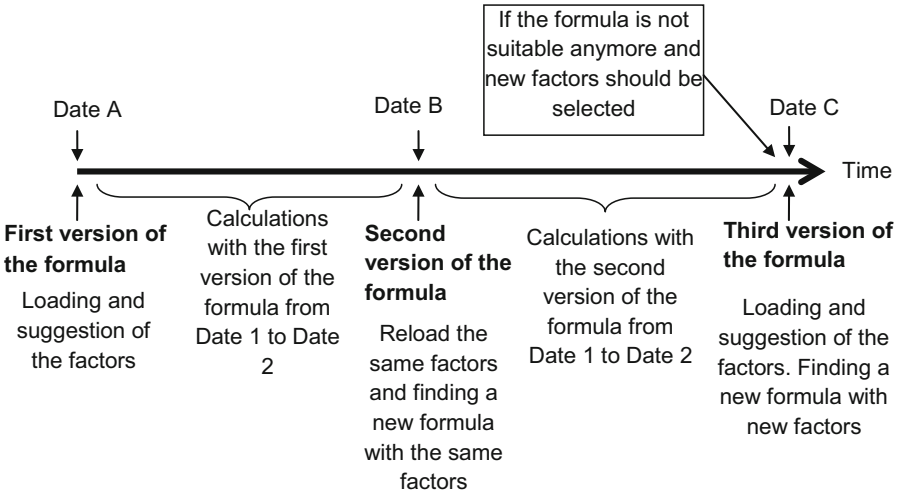


Fig. 3. Formula calibration

At first, target and explanatory factors are selected and loaded, creating the first version of the formula. This formula version is then applied to the calculations. After some time, when the generated target starts to deviate from the real values, the formula must be corrected. The selected explanatory factors, that have been used to build the first formula version, are applied again, but the formula is calibrated, new functions are selected and coefficients for the formula terms are produced. Thus, a second formula version is created, that is being used for the generation of the target until its values start again to deviate from the real target factor values. If this deviation is too significant, new explanatory factors should be selected. All factors are loaded again, a new factors selection is performed by one of the before mentioned automated approaches – clustering, min or max correlated – which can also be manually changed. The selected factors are used to generate the third formula version, where explanatory factors, basis functions and coefficients will be different in comparison to the previous formula versions. It can then be used for later calculations until a new calibration is needed.

Thus, there are two sorts of calibrations:

- Preserving currently selected explanatory factors and changing only the functions applied on them and on the regression coefficients, including the free term.
- Selecting new explanatory factors. In this case, new factors can be added, existing factors removed or both. The formula is being completely changed according to the basis functions and the regression coefficients.

In every formula calibration the settings can be changed, as the set of basis functions that can be used in the formula, with or without removing the terms with coefficients that are too small.

The experimental results show that the best results are obtained when the number of explanatory factors is close to, but not exceeding, the number of historical dates. It is

not quite clear which basis functions should be supplied in the evolutionary algorithm for finding the best possible modeling formula. That is one of the issues that should be investigated further. Nevertheless, the system has already been introduced in real financial software solutions and has been used for the purposes stated in the introduction.

4 The Multifactor Approach Applied to the ESG Rating Calculation

Figure 4 illustrates individual scorings of a sample company for each of the three main ESG factors – Environmental (39), Social (37) and Governance (54). The impact of each factor on the overall ESG rating, as well as on the balance between the three factors, can easily be derived from the figure.

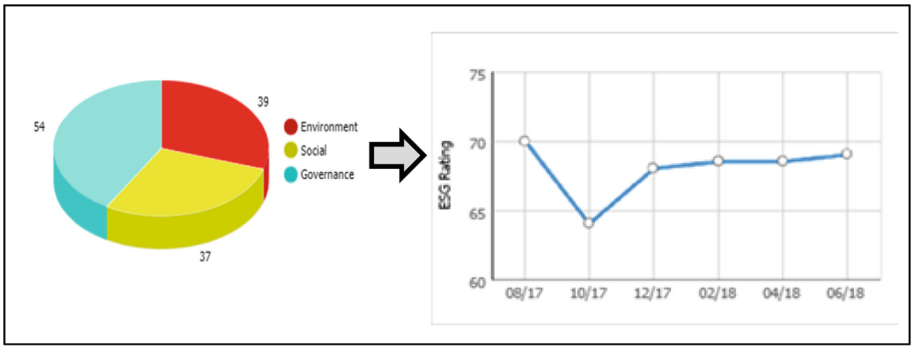


Fig. 4. Sustainability ESG categories and evolution of ESG rating

The ESG calculated ratings can be separated in two categories:

- Declarative rating – executed by the company itself. This rating is often considered subjective. It is not always clear how reliable it is, since companies tends to publish results that are in their own best interest.
- Requested rating – executed by rating providers on behalf of other companies. This rating can be considered as more reliable and it is in the focus of this research paper. Once developed, the methodology for an automatic rating evaluation can be used not only by rating providers, but also by any other company as well.

The presented approach represents an innovative, as well as objective and effective method for the evaluation of ESG ratings. Additionally, it saves human effort, which lessens the cost of a business organization and can advance its performance. One of its most important benefits, in contrast to current approaches, is the usage of objective market data (share prices, products prices, etc.), and not subjective company reported data. The data is comprised of time series of historical observations, which are used to construct a mathematical model that produces series that are as similar as possible to the target series. The target series contain public data observations of the company

under evaluation (the target company) and explanatory series are comprised of data observations of indices with known ESG scores. The relation is expressed as a formula, which generates synthetic series that come closest to the target series, when applied historically. Some explanatory factors in the formula participate with positive weights, while others do so with negative weights. For example, if companies have environmental or social results that are similar in behavior to the ones in the target company, they participate with positive weights in the mathematical expression. Contrary to that, if the weights are negative, companies demonstrate negative values for the corresponding evaluated factors. The process of mathematical modelling is illustrated in Fig. 5.

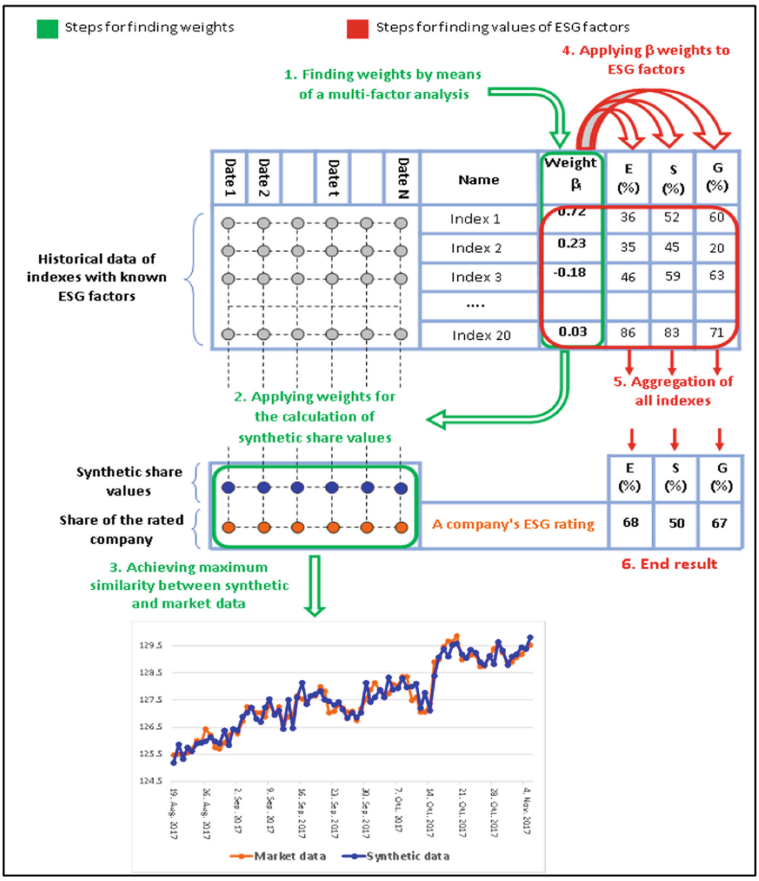


Fig. 5. Automated ESG rating calculation

First, weights β must be found in such a way that when applied to historical time series values, produce synthetic series that are as close as possible to the given target series of the target company. Once determined, these weights are applied to the known scores of factors E, S and G in order to produce estimations of unknown scores. Finally,

the three values are represented in the same way as the results in Fig. 4 and are used to calculate the overall ESG rating.

An important step in the proposed automated approach is the selection of a proper subset of indices with known ESG ratings. This must be completed before determining their influence on the target company, whether positive or negative. This subset can be defined manually or it can be achieved by using an automatic suggestion approach, or both. Peer companies can either be within the same domain as the target company or from different domains. They determine the positively and negatively weighted explanatory factors for the target company:

- **Positively weighted explanatory factors** – (Fig. 6) There is a positive correlation to the target company. For example, if the explanatory company has low CO_2 emissions, the target company too will have low CO_2 emissions.

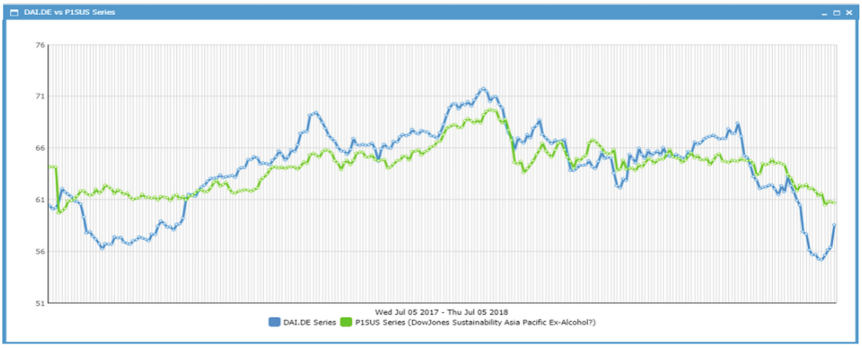


Fig. 6. Positive weight

Fig. 7. Negative weight

- **Negatively weighted explanatory factors** – (Fig. 7) There is a negative correlation to the target company. This means that a given factor of the explanatory company is in opposition to the same factor of the target company.

Once built, the mathematical model can be used for future ESG rating estimations. A calibration is performed only when the accuracy of the model is below a given threshold. An automatic calibration can be scheduled in preliminary determined time frames.

The usage of the automated approach does not exclude the possibility of working with the current subjective questionnaire based approach, or combining them together. Both approaches can be used simultaneously, each of them participating with different percentages to calculate the final ESG rating score. The Sustainability Evaluator also allows the questionnaire-based approach to be validated, as shown in Fig. 8 and according to [1].

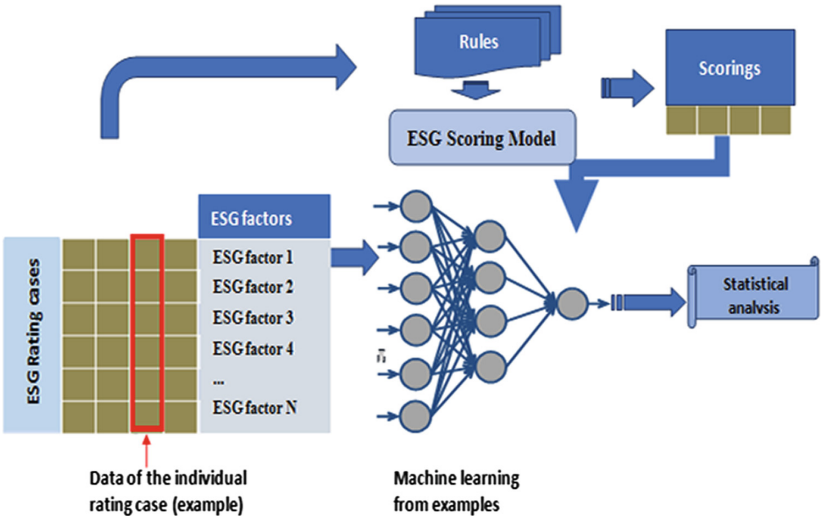


Fig. 8. Validation process of the subjective approach

The idea behind it is that the logic of the expert based questionnaire methodology is considered established and it is transferred to a neural network by training it with available examples. After that, the features of the system are evaluated by analyzing the neural network. Thus, the significance of indicators and certain internal concepts can be established.

Automated solutions that apply such artificial intelligence approaches have been widely used in the recent past due to improvements in hardware technologies and their increased effectiveness. Multi-factor modelling is already well-known in the financial sector and first experimental results have already demonstrated its usability when applied to the ESG rating evaluation in our solution. It automatically maps groups of data series into a target series, thereby creating a mathematically expressed relation between them and the target factor by a set of weights. The proposed automatic approach saves human effort and allows a more frequent ESG rating evaluation, compared to the traditional approach.

5 Conclusions and Future Work

Many companies use their own methodologies for ESG ratings evaluation, thereby providing analysis and additional information, such as compliance to standards and conventions, country ratings, etc. [4–6]. Some of the most active and most famous ESG rating providers worldwide are: Bloomberg (USA), MSCI (USA), Thomson Reuters (USA), Vigeo (France), EIRIS (UK), oekom (Germany), Inrate (Switzerland), Sustainalytics (Netherlands), Covalence (Switzerland), Corporate Governance Agency (Switzerland), Infras (Switzerland), SIRIS – Sustainable Investment Research Institute (Australia), CAER (Australia and New Zealand), Ecodes (Spain), Greeneye (Israel), IMUG (Germany) and KOCSR (South Korea), Trucost (UK), EthiFinance (France), Solaron (India), and others.

The Sustainability Evaluator can benefit a variety of market participants and has multiple advantages. First, investors will be better informed about available options they can choose in order to realize their investment strategies. ESG ratings help make informed decisions regarding a sustainable development of market participants. Second, the target company is also interested in improving its marketing policies by providing more information about non-financial ratings. Third, by paying attention to ESG ratings, companies are stimulated to improve their environmental, social and governmental activities, thus influencing and potentially improving the lives of their employees.

Even though ESG ratings do not represent financial information, they can be used by financial rating providers as additional data and can be applied to the assessment process of companies. ESG ratings are mainly used by financial institutions, credit rating agencies or in the insurance industry, but the Sustainability Evaluator can be integrated into any organization or industry that is engaged in production or providing services.

The majority of technical details concerning the Sustainability Evaluator have already been fully developed and were tested, while others are still in development. For example, the multi-factor methodology is used for other tasks as well, for instance, as a financial instrument for mapping and modelling the unknown content of indices. In time, it could also be used for multivariate prediction. The overall calculation of a single numeric ESG rating, generated from the scores of the three main factors, is also well-established. A demo software is available as a web application with various working functionalities. A validation methodology is being developed for financial credit ratings by a supervised trained neural network, which can be easily adapted to the Sustainability Evaluator.

Sustainability Evaluator enables any market participant to easily evaluate its own or other participants' ESG ratings. The solution can be integrated into other software systems; both web and desktop based and can work in different regimes.

Acknowledgments. This paper is supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)" (grant agreement DO1-205/23.11.18), financed by the Ministry of Education and Science.

References

1. Regulation (EU) No 575/2013 of the European Parliament and of the Council of 26 June 2013 on prudential requirements for credit institutions and investment firms and amending Regulation (EU) No 648/2012. https://publications.europa.eu/en/publication-detail/-/publication/ccd31733-df06–11e2-9165-01aa75ed71a1. Accessed 28 Sept 2019
2. 2017–2018 LG Electronics Sustainability Report. https://www.lg.com/global/sustainability/communications/sustainability-reports. Accessed 12 Sept 2019
3. ESG: Understanding the issues, the perspectives, and the path forward. https://www.pwc.com/us/en/services/governance-insights-center/library/esg-environmental-social-governance-reporting.html. Accessed 2 Sept 2019
4. MSCI ESG Ratings Methodology Executive Summary. MSCI ESG Research April 2018. https://www.msci.com/documents/10199/123a2b2b-1395–4aa2-a121-ea14de6d708a. Accessed 26 July 2019
5. Thomson Reuters ESG Scores. Date of issue: May 2018. http://zeerovery.nl/blogfiles/esg-scores-methodology.pdf. Accessed 11 Aug 2019
6. Novethic research: Overview of ESG rating agencies. https://www.novethic.com/fileadmin/user_upload/tx_ausynovethicetudes/pdf_complets/2013_overview_ESG_rating_agencies.pdf. Accessed 22 July 2019
7. Rosen, K.: Discrete Mathematics and Its Applications, 4th edn. AT&T (1998)
8. Steel, R., Torrie, J.: Principles and Procedures of Statistics. McGraw-Hill, New York (1960)
9. Cameron, C., Trivedi, P.: Regression Analysis of Count Data. Cambridge University Press, Cambridge (1998)
10. Koza, J.: Genetic Programming. MIT Press, Cambridge (1992)
11. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1999)
12. Draper, N., Smith, H.: Applied Regression Analysis. Wiley Series in Probability and Statistics. Wiley, New York (1998)
13. Hamilton, J.: Time Series Analysis. Princeton University Press, Princeton (1994)
14. Recktenwald, G.: Numerical Methods with Matlab: Implementations and Applications. Prentice Hall, Upper Saddle River (2007)
15. Nikolov, V., Naydenov, D.: Multifactor modelling system with cloud-based pre-processing. In: CompSysTech Proceedings of the 14th International Conference on Computer Systems and Technologies, ACM ICPS, NY, vol. 767, pp. 239–246. ACM Inc. (2013)

AI and Our Understanding of Intelligence

James P. H. Coleman(✉)

Edge Hill University, Ormskirk, Lancashire L39 3LG, UK
colemanj@edgehill.ac.uk

Abstract. Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans. It is a growing discipline that has enabled computer to undertake activities like facial recognition and game playing in a manner that, to an outsider, looks intelligent. There are many different toolsets that are called AI including neural networks, machine learning, expert systems, fuzzy logics, swarm intelligence and many others. These tools are being used in a wide variety of situations to solve a wide variety of complex problems, problems which have been difficult to solve using traditional algorithmic processes. Human intelligence and AI are sufficiently similar that tools developed to understand human intelligence can be applied to computer-based AI systems as well. Blooms Taxonomy is used to provide a definition of what constitutes intelligent computer applications. The article then considers the relationships between applications and intelligent systems.

Keywords: Artificial intelligence · Computational Intelligence · Bloom's Taxonomy

1 Introduction

1.1 Artificial Intelligence

Artificial Intelligence (AI) is defined as a set of "computer algorithms that are able to perform tasks normally requiring human intelligence" [1]. While this is a dictionary definition, other more practical definitions describe AI as being the study of intelligent agents where the agents "…perceives its environment and takes actions that maximize its chance of successfully achieving its goals" [2] while others describe AI-based systems as devices that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving" [3]. Equally importantly though, is developing systems (solutions) where the behaviours would be thought intelligent in human beings. It is reasonable to use this as a reference point for intelligence – that the behaviours exhibited by a computer system displays similar intelligent behaviour to humans.

AI techniques have become an important component of many IT systems, both in the Technology industry as well as in many ancillary industries including health and finance. AI helps to solve many challenging problems in computer science, software engineering and operations research [4].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 183–190, 2021.
https://doi.org/10.1007/978-3-030-55180-3_15

The phrase Intelligent is mostly used in the context of "…a person is intelligent…" or something similar which implies that it is a category of people, and there are people who are not intelligent. This is of course a nonsense statement because every person has intelligence, or to be more precise, every person displays behaviours that result from intelligence, so what is being discussed is the level or depth of intelligence behaviours.

The same can be applied to computer programmes – to what degree is a programme/application displaying intelligent behaviour, that is, to what extent is the application displaying artificial intelligence.

Human intelligence is a process of computation. Computation where the processor is a chemo-electrically based organism. Given this situation there is no reason to believe that the processing that occurs within the human body is fundamentally different from the processing that occurs within a computer-device. Further, we know that computers can model the operations of all individual elements of the human body provided we are able to understand the relationships between the different parts of the body, then we should be able to model human intelligence.

The fact that now, we do not fully understand how the human body works and interacts does not mean that these factors are unknowable, only that we do not know them yet. It is very clear that our understanding of human intelligence is very limited at the moment, but it is not clear that they are unknowable. Human intelligence can be used as a reference model for artificial intelligence, and therefore many of our understandings of human intelligence can be applied to artificial intelligence.

There are many behaviours that demonstrate intelligence, just as there are many different forms of intelligence. One tool that is used in human intelligence education to provide a framework to understand and develop intelligence is Blooms Taxonomy [5].

1.2 Bloom's Taxonomy

Bloom's Taxonomy was created by a committee chaired by Benjamin Bloom during the 1950s to categorize the types and levels of reasoning skills required in developing curricular material for classrooms. The Taxonomy (after revisions) identified three (3) separate domains:

- The cognitive domain (knowledge-based)
- The affective domain (emotion-based)
- The psychomotor domain (action-based)

The domain most useful for discussions here is that of the cognitive domain where there are 6 levels of abstraction in cognitive intelligence. While it was developed for the classroom environment, it can be used in other areas as it provides a framework for understanding intellectual skills, and the relationships between them. In this article, it will be used (with revisions) to describe the different types of intellectual skills used in AI.

The Cognitive Domain focusses on intellectual skills associated with cognition – the ability to think and make decisions based on rational thought and calculation. This domain is most relevant to modern computing where the systems are designed to process data.

The Affective Domain describes the way people react emotionally and their ability to empathise. This domain typically targets the awareness and growth in attitudes, emotion, and feelings.

The Psychomotor Domain describes the ability to physically manipulate a tool or instrument. Psychomotor objectives usually focus on change and/or development in behaviour and/or skills and is most relevant to robots and their development.

Parker J. (2016) [6] analyzed several different AI systems and discussed the extent to which these individual systems and their learning were comparable with Bloom's learning theory, the extent to which many AI systems demonstrated behaviours aligned to Bloom's. What is significant is not the extent that AI systems are able to learn, but that the comparison was even viable and a sensible comparison to make. Parker concluded that AI had not progressed very far in developing learning in AI systems, but this article showed that the author believed the comparison was valid and reasonable. The fact that AI is a very old discipline (over 50 years), it should not be surprising that the extent to which machine have developed intelligence is small given the time it has taken for humans to develop intelligence and being able to understand that intelligence has taken many more years.

The six levels of the Cognitive Domain are:

- Knowledge - Remember - Knowledge involves recognizing or remembering facts or data without necessarily understanding what they mean.
- Comprehension - Understand - understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions.
- Application - Apply - using acquired knowledge—solving problems in new situations by applying acquired knowledge, facts, techniques and rules.
- Analysis - Analyze - drawing connections among ideas, examining and breaking information into component parts, determining how the parts relate to one another, identifying motives or causes, making inferences, and finding evidence to support generalizations.
- Evaluation - Evaluate - Justify a stand or decision, presenting and defending opinions by making judgments about information, the validity of ideas, or quality of work based on a set of criteria.
- Synthesis – Create - Produce new or original work.

When an AI tool can display the behaviours associated with the different categories in Bloom, then it is reasonable to say that the tools are behaving at that level. This then describes the types of intelligence being exercised.

For the Affective Domain (emotion-based), the five levels are:

- Receiving - The lowest level; the system is able to receive data and store that data.
- Responding - The system actively participates in the process, not only attends to a stimulus; the student also reacts in some way.
- Valuing - The system attaches a value to an object, phenomenon, or piece of information. The system associates a value or some values to the knowledge acquired. The data has a purpose within the system.

- Organizing - The system can put together different values, information, and ideas, and can accommodate them within his/her own schema; the system is comparing, relating and elaborating on what has been learned.
- Characterizing - The system at this level is able to build abstract knowledge.

For the Psychomotor Domain (action-based), the six levels as developed by Simpson [7] are:

- Perception - The ability to use sensory cues to guide motor activity: This ranges from sensory stimulation, through cue selection, to translation.
- Set - Readiness to act: It includes mental, physical, and emotional sets. These three sets are dispositions that predetermine a person's response to different situations (sometimes called mindsets). This subdivision of psychomotor is closely related with the "responding to phenomena" subdivision of the affective domain.

  – Examples: Knows and acts upon a sequence of steps in a manufacturing process. Recognizes his or her abilities and limitations. Shows desire to learn a new process (motivation).

- Guided response - The early stages of learning a complex skill that includes imitation and trial and error: Adequacy of performance is achieved by practicing.
- Mechanism - The intermediate stage in learning a complex skill: Learned responses have become habitual and the movements can be performed with some confidence and proficiency.
- Complex overt response - The skillful performance of motor acts that involve complex movement patterns: Proficiency is indicated by a quick, accurate, and highly coordinated performance, requiring a minimum of energy. This category includes performing without hesitation and automatic performance. For example, players will often utter sounds of satisfaction or expletives as soon as they hit a tennis ball or throw a football because they can tell by the feel of the act what the result will produce.
- Adaptation - Skills are well developed and the individual can modify movement patterns to fit special requirements.
- Origination - Creating new movement patterns to fit a particular situation or specific problem: Learning outcomes emphasize creativity based upon highly developed skills.

While Bloom and his colleagues created the levels for the Cognitive and Affective Domains, no such levels were created for the Psychomotor Domain. Several research groups developed sets of levels for this domain. The one used here is the system developed by Simpson in 1972.

2 Human and Biological Intelligence

Human Intelligence (HI) is the only model we have for a developed intelligence system that incorporates all levels of Bloom's Taxonomy. Human intelligence incorporates many Biological Intelligent (BI) Systems, with many examples being in the Psychomotor and

Affective Domains, but not exclusively. Examples of such intelligence systems include such systems as knee-jerk/reflex reactions, breathing etc. which operate independently of human management.

Owens [8, 9] discusses the situation of the human body processing visual data even while in a "vegetative state". While the research focused on using functional neuroimaging to detect awareness in patients who are incapable of generating any recognized behavioral response and appear to be in a vegetative state, it also illustrates that the human body process visual data (and other data types) without direct conscious action on the part of the human concerned. That is, the visual data is processed by the brain automatically because light enters the eyes and the light sensors at the back of the eye – this research shows that there are a number of intelligent systems that are operating in parallel, and that it is up to other supervisory controls in the brain to actually take notice of the results of the data processing and make decisions about how to react to the visual input and its meaning.

This situation demonstrates that within the human body, there are multiple independently intelligent systems that co-operate to demonstrate behaviour we view as being intelligent.

BI is intelligence that enables specific biological processes to be undertaken in an intelligent fashion. For example, human breathing occurs at a rate that ensures the body receives sufficient oxygen to undertake its operations. Adapting the rate and depth of breathing to suit needs of the body. So a person that is short of oxygen will breathe harder, while a body that is unable to gain sufficient oxygen will change the operation of non-core bodily organs (resulting in create "aches/pains" in the body) to try and force the body to reduce its need for oxygen. This is done in such a fashion that the body hardly ever receives too much or too little oxygen.

Some intelligent systems (IS) (used by the human body) are automatically processed by the body. Possible examples include such features as the ability to process visual data, process audio data, the human reflex as seen with the application of heat to the body etc. It is sensible to propose that many body functions that are considered to be under conscious control work in a similar manner. So for example, it is consistent with our understanding of how our mind processes mathematical processing, that there are specialist services that support and manage our processing of mathematical concepts. So for example, a child who does not understand mathematics will observe the existence of pictograms:

$$2 + 3 = 5 \text{ or } a + b = c$$

however, only people who have learnt about mathematics and the Latin character sets would view this same sequence of characters as being an addition operation. Similarly, the same example would be applicable for examples written in Arabic or south-east Asian characters sets where someone who does not understand the characters being used could be confused by such character patterns.

Anderson [10] from Carnegie Mellon University and his team identified four distinct stages to the process of understanding language/mathematics: encoding (reading and understanding the problem); planning (working out how to tackle it); solving (crunching the numbers); and responding (typing in the correct answer).

This process suggests that for this process, there are four (4) distinct stages to the processing of mathematical problems. Experience of looking at how the body processes

bodily support systems (e.g. breathing) would show that each stage may involve different systems that cooperate. This implies that there is also a separate supervisory layer that coordinates the operations of the separate systems – the encoding, planning, solving and responding (see Fig. 1). This model places the Supervisor as the system that coordinates the four (potentially) independent activities.

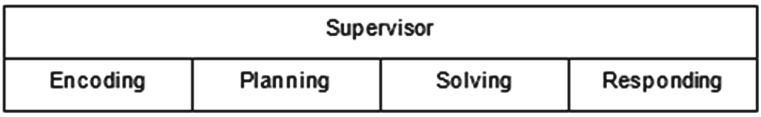


Fig. 1. Diagram showing the relationship between separate intelligent systems.

Owen [8] demonstrates that the human intelligence is a system of integrated (and in many cases in parallel, not just in sequence) components rather than a single application that produces intelligent behaviour. So, for example, the operation of the organs of the human body affect aspects of human behaviour. A common example is that of pain which often results in a person being short-tempered, affecting their thinking processes and therefore their behaviour.

2.1 Intelligence and Applications

Computational Intelligence (CI) is a subset of AI and relates to the theory, design, application and development of biologically and linguistically motivated computational paradigms. The main pillars of CI have been Neural Networks, Fuzzy Systems and Evolutionary Computation. However, in time other nature-inspired computing paradigms have evolved. Thus, CI is an evolving field and at present in addition to the three main constituents, it encompasses computing paradigms like ambient intelligence, artificial life, cultural learning, artificial endocrine networks, social reasoning, and artificial hormone networks.

Engelbrecht [11] defines CI as "the study of adaptive mechanisms to enable or facilitate intelligent behaviour in complex and changing environments. Computational intelligence is certainly more than just the study of the design of intelligent agents, it includes also study of all non-algorithmizable processes that humans (and sometimes animals) can solve with various degree of competence [12].

CI sub-field is defined by the author Andries P. Engelbrecht [13] as the study of 'adaptive mechanisms' which enable or facilitate intelligent behaviour in complex and changing environments. In other words, CI is mainly about the design of algorithmic models to solve complex problems. The four paradigms are, in order of presentation: artificial neural networks; evolutionary computing; swarm intelligence and fuzzy systems.

CI is characterized by focusing on methods and tools that are not the traditional programming paradigms ed OO Programming or procedural language programming because the problem domain has not been found to be conducive to providing solutions. Systems like image recognition or natural language processing. These paradigms are not considered to be AI systems by the IT industry, but that does not, in any way, alter the fact that the systems demonstrate behaviours that are identifiable from within Bloom's

Taxonomy – that is, they are intelligent systems, but they are systems that generally are not identified as being intelligent.

While this attitude is a valid approach in the general life of application development and operation, they do meet the description of behaviour that Bloom's Taxonomy identifies, therefore it is reasonable to view them as being intelligent systems, though the system may only be low-level intelligence.

If we adopt the definition that an intelligent system is a system that manipulate facts and concepts and demonstrates behaviours that are in accord with the different levels of Blooms taxonomy, some of which must be above the basic level (e.g. knowledge for the Cognitive Domain).

> **Definition.** An intelligent system is a system that manipulate facts and concepts and demonstrates behaviours that are in accord with the different sub-levels of Blooms taxonomy, at least one of which must be above the basic level in the related Domains.

With this definition, applications that only store and retrieve data would not be classed as being intelligent – for example, a simple daily diary would not fall under this definition. This meets the current industry understanding of what an intelligent system is. On the other hand, it would be fair to describe a relational database application (RDBMS) as an intelligent application because it has the capacity to organize data, comparing data against other criteria and possibly giving descriptions. This is at level 2 (comprehension level of cognitive domain) and therefore make the RDBMS an intelligent application. While this does not align with common understanding, perhaps we should re-consider relational database management systems as being intelligent systems.

The Taxonomy provides a powerful framework for describing behaviours, not just intellectually, but also involving physical movement. The full set of the Taxonomy's domains would be applicable for a traditionally conceived "robot" where the android needs to react in the human world, and not just in the current "computing world" and needs to make decisions about future goals. A robot walking across a street or an unmanned motor vehicle driving down a public street would need to incorporate aspects of the psychomotor domain. Examples could include: an unmanned car overtaking another vehicle or a robot crossing a street would need to judge whether it is safe to cross a street.

A robot career, on the other hand, would need to be able to use skills from the affective domain when caring for ill or sick patients and be able to use these skills to provide appropriate care. They would also be useful for therapy-bots who work with patients in providing spoken-therapy.

Unlike the many definitions of machine intelligence discussed in [14], this approach has the advantage of being based on sound cognitive understanding of intelligence (Bloom's Taxonomy), while still proving to be adaptable to machine intelligence and artificial intelligence. What it shows is that we are only very early in the intelligence development cycle, and that there is a long way to go.

3 Conclusion and Future Research

Bloom's Taxonomy provides us with a set of tools that enables developers and researchers to understand the intelligence that is demonstrated by the applications that are being developed. This toolset provides a scale that can be used to *measure* the type and extent of intelligent behaviour that a system displays.

Further, in describing the process of describing the process of processing natural language, we have recognized an important schema for future intelligent systems – the role of the *supervisor* in managing and processing other intelligent systems gives us a layout for developing future intelligent systems.

The toolset of Bloom's Taxonomy will need further refinement to ensure that it continues to meet the needs pf developers across the full range of IT systems particularly as we process up the taxonomical pyramid. As has been seen in this work, we are still very in the Bloom's Taxonomy in our development of artificial intelligence. As more intelligent behaviour is developed, it will be important to ensure that our measuring tool remains relevant.

References

1. Dictionary.com. https://www.lexico.com/en/definition/artificial_intelligence. Accessed 10 Oct 2019
2. Poole, D., Mackworth, A., Goebel, R.: Computational Intelligence: A Logical Approach. Oxford University Press, New York (1998). ISBN 978-0-19-510270-3
3. Russell, S., Norvig, P.: Artificial Intelligence a Modern Approach, 3rd edn. Prentice Hall, Upper Saddle River (2009)
4. Clark, J.: Why 2015 Was a Breakthrough Year in Artificial Intelligence. Bloomberg News 8 December 2015. Accessed 10 Oct 2019
5. Bloom, B.S. (ed.). Taxonomy of Educational Objectives. Cognitive Domain, vol. 1. McKay, New York (1956)
6. Parker, J., Jaeger, S.: Learning in Artificial Intelligence: Does Bloom's Taxonomy Apply? (2016)
7. Simpson, E.: Educational Objectives in the Psychomotor Domain. Gryphon House, Washington, D.C. (1972)
8. Owens, A: "The Life Scientific" 22/10/2019 on BBC Radio4. https://www.bbc.co.uk/sounds/play/m0009ksc. Accessed 10 Oct 2019
9. Owen, A., Coleman, M.: Detecting awareness in the vegetative state. Ann. N. Y. Acad. Sci. **1129**, 130–138 (2008)
10. Anderson, J.R., Pyke, A.A., Fincham, J.M.: Hidden stages of cognition revealed in patterns of brain activation. Psychol. Sci. **27**(9), 1215–1226 (2016). https://doi.org/10.1177/0956797616654912
11. Engelbrecht, A.: Computational Intelligence: An Introduction. Wiley, New York (2003)
12. Duch, W.: What is computational intelligence and where is it going? In: Challenges for Computational Intelligence, pp. 1–13. Springer, Heidelberg (2007)
13. Engelbrecht, A.: Computational Intelligence: An Introduction. Wiley, New York (2002)
14. Legg, S., Hutter, M.: Universal intelligence: a definition of machine intelligence. Minds Mach. **17**(4), 391–444 (2007)

Fault Diagnosis and Fault-Tolerant Control for Avionic Systems

Silvio Simani[1(✉)], Paolo Castaldi[2], and Saverio Farsoni[1]

[1] Department of Engineering, University of Ferrara, Ferrara, Italy
silvio.simani@unife.it
[2] Department of Electrical, Electronic, and Information Engineering, University of Bologna, Bologna, Italy
http://www.silviosimani.it

Abstract. This paper addresses the development of an active fault tolerant control scheme for avionic systems. The methodology is applied to an aircraft longitudinal autopilot taking into account possible faults on the aircraft actuators. The key feature of the proposed control relies on its active characteristics, as the fault diagnosis strategy is based on a robust estimate of the fault signals that are compensated. The design method uses an intelligent data–driven scheme via a fuzzy modelling and identification procedure, which derives these adaptive filters with disturbance decoupling features. The work shows that these fault estimates can be used for fault accommodation. In particular, the fuzzy approach proposed in the paper provides the reconstruction of the fault signals that are decoupled from the wind components, and thus applied to the aircraft system. The proposed solutions provide interesting robustness features that are analysed by using a high–fidelity simulator, which is able to include different operating points and realistic actuator faults, turbulence, measurement errors, and the model–reality mismatch.

Keywords: Fault diagnosis · Fault tolerant control · Fault estimation · Fuzzy systems · Data-driven approach · Avionic system

1 Introduction

In general, a feedback control design for complex and safety–critical systems can achieve unsatisfactory performance, and possible instability, especially in presence of faults affecting actuators, sensors, and system components. This is true for aircraft and spacecraft applications, as considered in this work. For these processes, even the effect of an incipient fault [10] in a system component, actuator or sensor, can generate catastrophic consequences.

To this end, Fault Tolerant Control (FTC) solutions have been proposed to compensate component faults, while coping with desirable stability, and graceful performance degradation [10].

In general, FTC schemes can be divided into two types, *i.e.* Passive FTC Scheme (PFTCS), and Active FTC Scheme (AFTCS) [12]. This definition relies

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 191–201, 2021.
https://doi.org/10.1007/978-3-030-55180-3_16

on the capabilities of the designed controller to be robust against a class of presumed faults (PFTCS) or to actively accommodate the control law, so that stability and acceptable performance of the entire system can be maintained (AFTCS). The latter solution relies heavily on the Fault Detection and Diagnosis (FDD) task, which is able to provide update information about the faulty conditions. The strategy proposed in this work relies on an AFTCS.

Many FDD techniques have been developed, see *e.g.* [5,6]. Regarding the AFTCS design, it is clear that an effective FDD is required [3,4]. Moreover, the system can react properly and promptly to a fault, if an accurate FDD scheme is implemented. FDD solutions represent a challenging point as they have to provide a reliable and robust fault estimate, as discussed *e.g.* in [7].

This paper proposes an Artificial Intelligence (AI) method for the design of an AFTCS applied to an aerospace process. The designed FDD module, used for its fault estimation capabilities, is computed using a fuzzy approach. Hence, the obtained solution is applied to the aircraft longitudinal system, in order to achieve fault estimates insensitive to disturbance and other faults. Moreover, this paper considers actuator faults, whose estimations accomplished via fuzzy prototypes are decoupled from the considered wind components.

The key point of the study relies on the proposed FDD strategy, which increases the reliability and the robustness of the complete AFTCS. The overall AFTCS is designed by including an inner feedback loop to the baseline controller, which compensate the fault effects. Another important issue is that the controller, which was previously designed on the nominal fault–free plant, does not need to be replaced. Moreover, the overall AFTCS is able to maintain the stability properties of the original system, and to limit the performance degradation of the nominal controller.

The work is organised as follows. Section 3 revises the scheme of the proposed AFTCS. Its design achieved via fuzzy filter, for fault estimation and accommodation is also addressed. On the other hand, Sect. 2 gives more details on the flight simulator, whilst Sect. 4 illustrates the efficacy of the developed AFTCS strategy by extensive simulations, which highlight its reliability and robustness characteristics. Concluding remarks and open problems for further research are finally drawn in Sect. 5.

2 Aircraft Simulator

The simulated aircraft models a Piper PA–30, which includes high–fidelity descriptions of its aircraft dynamics, propeller aerodynamics, and engine. They were implemented for simulation purposes, as described in [5]. The diagram of the aircraft system is summarised in Fig. 1, whilst its mathematical expressions are summarised below, in order to enhance the interested reader.

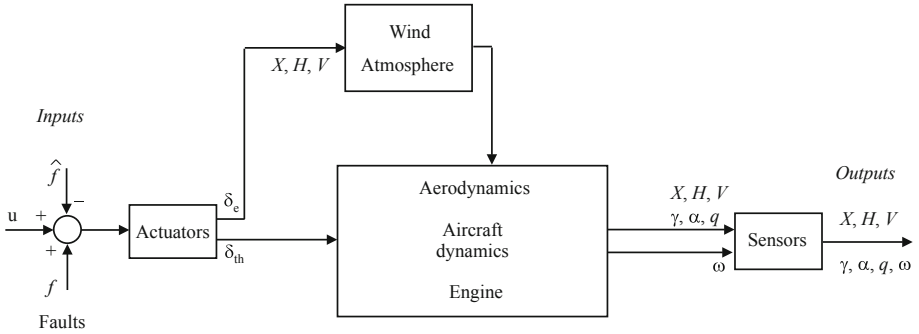


Fig. 1. Sketch of the aircraft simulator.

The aircraft mathematical description is given by the relations of Eq. (1) in a compact form [8]:

$$\begin{cases} \dot{X} = V\cos\gamma + U^w \\ \dot{H} = V\sin\gamma - W^w \\ \dot{V} = \frac{1}{m}\left[T\cos\alpha - D - mg\sin\gamma\right] + V\sin\gamma\cos\gamma\frac{\partial W^w}{\partial X} \\ \dot{\gamma} = \frac{1}{mV}\left[T\sin\alpha + L - mg\cos\gamma\right] + \cos^2\gamma\frac{\partial W^w}{\partial X} \\ \dot{\alpha} = q - \dot{\gamma} \\ \dot{q} = \frac{d_T}{I_y}T + \frac{M}{I_y} \end{cases} \tag{1}$$

The aircraft model parameters are reported in Table 1.

Table 1. The main aircraft model parameters.

Parameter	Variable
X	X inertial coordinate
H	Altitude
V	Airspeed
γ	Air ramp angle
U^w	Horizontal wind
W^w	Vertical wind
m	Aircraft mass
α	Attack angle
g	Gravity constant
q	Body pitch rate
d_T	Thrust arm
I_y	y-inertia momentum

Their relations are described by the expressions of Eq. 2:

$$\begin{cases} D &= \frac{1}{2}\rho V^2 S C_D \\ L &= \frac{1}{2}\rho V^2 S C_L \\ M &= \frac{1}{2}\rho V^2 S \bar{c} C_m \\ C_D &= C_{D_0} + C_{D_\alpha}\alpha \\ C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{\bar{c}}{2V}\left(q + \frac{\partial W^w}{\partial X}\right) + C_{L_{\delta_e}}\delta_e \\ C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{\bar{c}}{2V}\left(q + \frac{\partial W^w}{\partial X}\right) + C_{m_{\delta_e}}\delta_e \end{cases} \tag{2}$$

The aircraft aerodynamics and its coefficients included in Fig. 1 are recalled in Table 2.

Table 2. Aircraft model aerodynamic parameters.

Parameter	Variable
D	Drag force
L	Lift force
M	Pitch momentum
ρ	Air density
$\bar{c}$	Mean aerodynamic chord
$C_{D_\#}$	Drag coefficients
$C_{L_\#}$	Lift coefficients
$C_{m_\#}$	Momentum coefficients
δ_e	Elevator

The mathematical description of the engine of the aircraft simulator has the form of Eq. (3), whose variables are reported in Table 3:

$$\begin{cases} \dot{\omega} &= -\frac{Q_f}{I} + \frac{P_E}{I\omega} - \frac{P_P}{I\omega} \\ P_E &= C_1 H + C_2\omega\left[\delta_{th}\left(C_3 - C_4 H\right) - C_5\right] \\ Q_f &= J_v\omega^3 \end{cases} \tag{3}$$

Table 3. Aircraft engine model parameters.

Parameter	Variable
ω	Angular rate
Q_f	Friction torque
P_E	Power
I	Inertia
$C_\#$	Various coefficients
δ_{th}	Throttle
J_v	Friction coefficient

Finally, the aircraft propeller has the mathematical form of Eqs. 4, whose parameters are summarised in Table 4:

$$\begin{cases} P_P = C_P \left(\dfrac{V \cos \alpha}{\omega D} \right) \rho \omega^3 D_{PR}^5 \\ T \ = \dfrac{2}{V \cos \alpha} \eta \left(\dfrac{V \cos \alpha}{\omega D} \right) P_P \end{cases} \tag{4}$$

Table 4. Propeller model of the aircraft system.

Parameter	Variable
P_P	Propeller Power
T	Thrust
C_P	Prop. power coefficient
D_{PR}	Propeller diameter
η	Propeller efficiency

The wind gusts included in the aircraft simulator are provided by the discrete wind gust block available in the Matlab and Simulink environments. Also the Dryden turbulence model (translational and rotational) has been considered by means of the Aerospace Blockset in the Matlab environment. Figure 1 highlights that the aircraft simulator includes the model of the measurements system [5,8].

Finally, the aircraft simulator includes an altitude and airspeed autopilot developed in [7] for the fault–free system. It was shown that this controller guaranteed the local asymptotic Lyapunov stability for any cruise equilibrium point.

3 Active Fault Tolerant Control System (AFTCS)

Figure 2 describes the accommodation logic implemented by the designed AFTCS. The signal u_r represents the reference input, whilst u is the actuated input. The signal y is the measured output, with f the actuator fault, and $\hat{f}$ its estimated signal.

To this aim, the scheme of Fig. 2 highlights that the proposed AFTCS integrates the designed FDD module that compensates the baseline controller. The estimated signal provided by the FDD module is injected into the inner control loop, thus compensating the effect of the actuator fault.

This FDD module relies on a bank of fuzzy filters and it is able to asymptotically provide the accurate fault estimation. Moreover, an uniform rate of convergence is also guaranteed, as shown in [6].

It is worth noting that, on the basis of the separation principle, the baseline controller can be directly designed for the fault–free and nominal aircraft model, thus representing an important benefit and one of the key issues of the proposed fault tolerance solution.

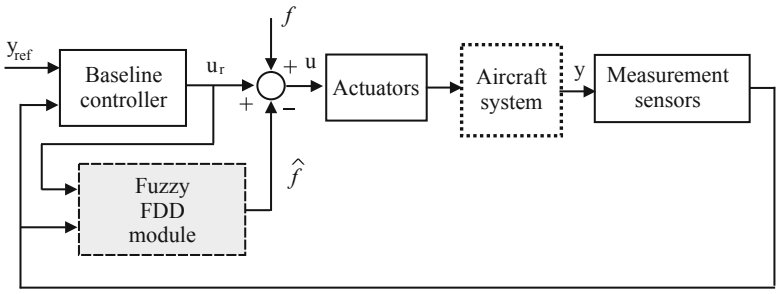


Fig. 2. Accommodation scheme of the proposed AFTCS.

The fault diagnosis scheme, *i.e.* the FDD module of Fig. 2 is designed by considering a NARX structure, thus exploited for providing an accurate reconstruction of the fault signal f. This study assumes that the considered aircraft plant is affected by additive faults affecting its input and output measurements. Moreover, measurement errors are also included, as described by Eqs. (5):

$$\begin{cases} u(k) = u^*(k) + \tilde{u}(k) + f_u(k) \\ y(k) = y^*(k) + \tilde{y}(k) + f_y(k) \end{cases} \tag{5}$$

with $u^*(k)$ and $y^*(k)$ the process variables, whilst $u(k)$ and $y(k)$ are the measurements acquired from the sensors, and $\tilde{u}(k)$ and $\tilde{y}(k)$ represent their measurement errors. Moreover, Eqs. (5) take into account the fault $f_u(k)$ and $f_y(k)$ signals, which have equivalent additive effects. In general, a number of r inputs and m outputs is considered.

The FDD scheme consisting of a bank of fault estimators is sketched in in Fig. 3. This study exploited this strategy as it is able to provide also the fault isolation tasks, as shown in [2].

Figure 3 highlights that in general the fault estimators use the input and output measurements $u(k)$ and $y(k)$. Moreover, the number of outputs of the FDD module is equal to the overall number of control inputs and monitored outputs, *i.e.* $r+m$. This solution allows also to solve the fault isolation task in a simple and straightforward way. The FDD module generates the fault estimation signals f_u and f_y.

On the other hand, the inputs of the fault estimators of the FDD module are selected via a fault sensitivity procedure. For each case, the fault signals and their effects on the aircraft system are considered. In particular, the most sensitive input $u_j(k)$ and output $y_l(k)$ measurements with respect to the considered fault are selected. In this way, fuzzy estimators include only the most effective input–output measurements, $u_j(k)$ and $y_l(k)$, which are used to reconstruct the fault signals $f_i(k)$, as highlighted by Fig. 3.

It is worth noting that this configuration allows to isolate multiple faults occurring at the same time.

The following of this section recalls the design of the fault estimators for the FDD bank obtained by means of Takagi–Sugeno (TS) prototypes. In this way,

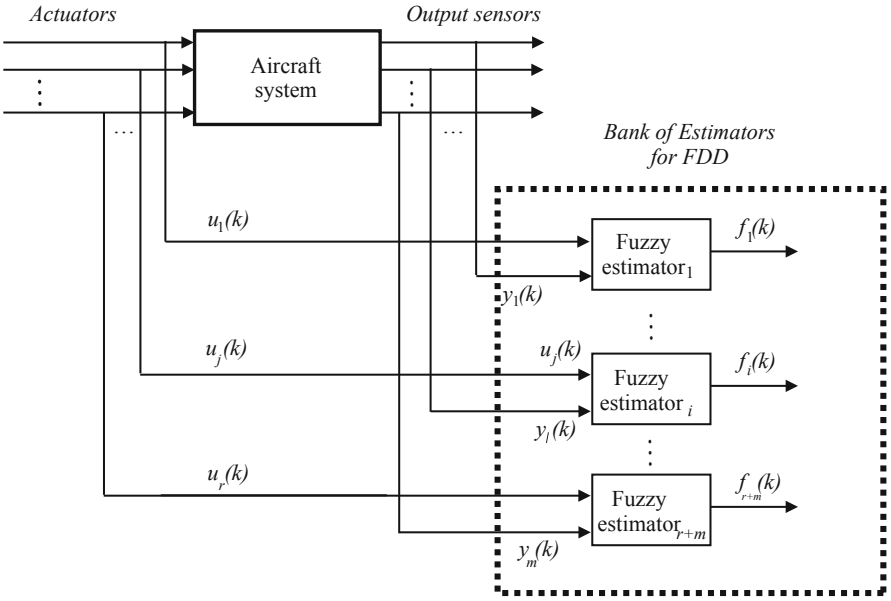


Fig. 3. Estimator bank for FDD and AFTCS.

the unknown relationships between the input–output measurements of the FDD bank of Fig. 3 and the considered faults are represented via fuzzy structures [1]. These structures are implemented by means of a Fuzzy Inference System (FIS) developed in the Matlab and Simulink environments [1].

According to this approach, the estimators of the FDD bank are represented as a set of nonlinear Multi–Input Single–Output (MISO) filters in the form of Takagi–Sugeno (TS) fuzzy structures [1]. The TS prototype considered here implements the consequents as deterministic functions $g_i(\cdot)$ of the inputs, while the antecedents remain fuzzy propositions represented by fuzzy sets and rules.

The i–th fuzzy rule of the FIS is described by the relation of Eq. (6):

$$R_i : IF \quad \text{(fuzzy combination of inputs)} \quad THEN \quad \text{output} = g_i(\text{inputs}) \tag{6}$$

with i representing the i–th rule. The antecedents are represented by membership functions $\lambda_i(x)$, which are activated by the input and output signals via proper linguistic propositions [1]. The consequent function $g_i(\cdot)$ in each rule R_i is represented by the piecewise affine function of Eq. (7):

$$g_i(x) = a_i^T x + b_i \tag{7}$$

with a_i and b_i its parameter vector and scalar offset, respectively. The number of rules R_i equals the number of clusters n_C that provide the partitioning of the data into regions (*i.e.* the rules R_i) where the relations $g_i(\cdot)$ are valid [1]. Furthermore, the antecedent of each rule depends on the membership function

$\lambda_i(x)$. Therefore, the overall TS fuzzy model is expressed as fuzzy superposition of the parametric models $g_i(x)$.

When the TS prototype is considered as generic fuzzy estimator, it has the form of Eq. (8):

$$\hat{f} = \frac{\sum_{i=1}^{n_C} \lambda_i(x)\, g_i(x)}{\sum_{i=1}^{n_C} \lambda_i(x)} \tag{8}$$

It is worth noting that the input vector x of the TS model of Eq. (8) contains the current as well as delayed samples of the system input and output signals. In this way, the dynamics are included into the static relation of Eq. (6), and the consequents represent discrete–time linear AutoRegressive descriptions with eXogenous input (ARX) of order o. According to this description, its regressor vector has the form of Eq. (9):

$$\begin{aligned} x(k) = \big[&\ldots, y_l(k-1), \ldots, y_l(k-o), \ldots \\ &\ldots u_j(k), \ldots, u_j(k-o), \ldots \big]^T \end{aligned} \tag{9}$$

where $u_l(\cdot)$ and $y_j(\cdot)$ are the aircraft input and output measurements $u(k)$ and $y(k)$ selected via the scheme represented in Fig. 3. The variable k refers to the time step, with $k = 1, 2, \ldots, N$. The parameters of i–th model of the Eq. (7) are represented by the vectors a_i:

$$a_i = \left[\alpha_1^{(i)}, \ldots, \alpha_o^{(i)}, \delta_1^{(i)}, \ldots, \delta_o^{(i)} \right]^T \tag{10}$$

with $\alpha_j^{(i)}$ referred to the output samples, whilst $\delta_j^{(i)}$ to the input ones.

This work proposes to derive the FIS for the design of the FDD scheme by using a system identification approach from the noisy data. According to this approach, the derivation of the parameters a_i and b_i in Eq. (7) is achieved via the methodology developed by the authors in [9]. This strategy was based on the optimisation of the prediction errors provided by the TS fuzzy models, and recast into an optimal estimation problems. The solution required the Errors–In–Variables description [9], which represents also the assumption of Eqs. (5).

A final key point, which is remarked here, concerns the derivation of the optimal number of clusters n_C. This issue is included in the estimation procedure developed by the authors in [11], which provided also the antecedent degrees of fulfilment μ_{ik} required in Eq. (8).

4 Simulation Results

This section shows the achieved results with reference to the determination of the FDD filters and the design of the overall AFTCS strategy. In particular, in order to define the fuzzy filter of Fig. 3 for the estimation of the elevator fault $\hat{f}_{\delta_e}$, the methodology recalled in Sect. 3 has been considered.

In this way, the fuzzy filter decoupled from both the aerodynamic disturbance d_j and the other fault, *i.e.* the throttle $f_{\delta_{th}}$, has been obtained.

With the same procedure, the fuzzy filter for the estimation of the throttle fault, $f_{\delta_{th}}$, and decoupled from the other fault, *i.e.* the elevator f_{δ_e}, and the wind gusts d_j, has been determined.

As an example, Fig. 4 (a) represents the fault on the elevator δ_e (black dotted line), $f_{\delta_e} = 1°$. It is compared with its estimation (black bold line) during the phase of altitude hold flight.

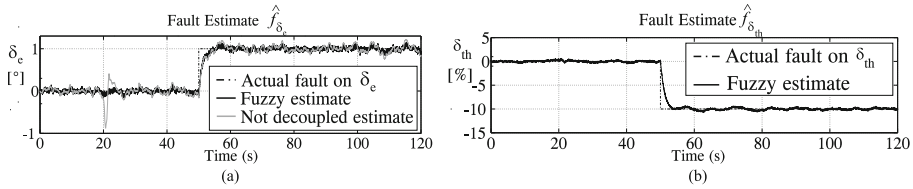


Fig. 4. (a) Fault f_{δ_e} and (b) $f_{\delta_{th}}$ estimates.

Figure 4 (a) highlights that the fault is reconstructed with a time delay smaller than the characteristic flight dynamics period. Moreover, its estimate $\hat{f}_{\delta_e}$ converges to the actual fault, which commences at $t = 50\,\text{s}$. Moreover, by comparing the estimate provided by the filter that is not decoupled from the disturbances, Fig. 4 (a) shows also that the fault estimation $\hat{f}_{\delta_e}$ does not depend on the wind disturbances. In fact, without any disturbance decoupling feature, the wind gust at $t = 20\,\text{s}$ would have affected the signal $\hat{f}_{\delta_e}$ (grey bold line). On the other hand, Fig. 4 (b) shows the fault $f_{\delta_{th}} = -10\%$ affecting the throttle δ_{th} and its estimation $\hat{f}_{\delta_{th}}$.

The remainder of this section shows the evaluation of the performance for the proposed AFTCS applied to the considered aircraft system. The simulations of the controlled aircraft with or without the designed AFTCS are compared.

As in the previous cases, the simulations included the presence of wind gusts and measurement errors, which have served to verify and validate the proposed solutions. If the fault has been timely detected and accurately estimated, the fault–free conditions can be maintained with graceful performance degradation. To this aim, Fig. 5 (a) shows the effects of both the wind gust and the elevator fault f_{δ_e} on the altitude controlled variable H.

On the other hand, Fig. 5 (b) shows the aircraft variable V in case of fault $f_{\delta_{th}}$. Note finally that the achieved results refer to the same conditions and highlight the efficacy of the proposed AFTCS strategy, in terms of robustness and reliability of the developed tools.

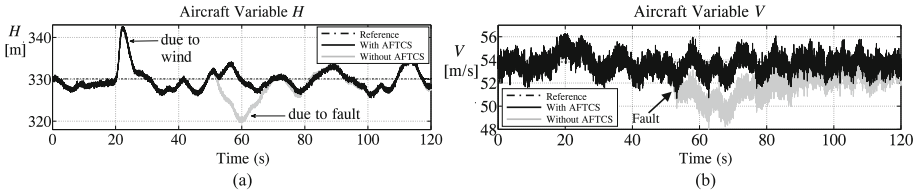


Fig. 5. (a) Aircraft altitude H and (b) speed V with and without AFTCS.

5 Conclusion

This paper addressed the development of an active fault tolerant control scheme for avionic systems. The methodology was applied to an aircraft longitudinal autopilot taking into account possible faults on the aircraft actuators. The key feature of the proposed control relied on its active characteristics, as the fault diagnosis strategy was based on a robust estimate of the fault signals that were thus compensated. The design method used an intelligent data–driven scheme via a fuzzy modelling and identification procedure, which derived these adaptive filters with disturbance decoupling features. The work showed also that these fault estimates can be used for fault accommodation. In particular, the fuzzy approach proposed in the paper provided the reconstruction of the fault signals that were decoupled from the wind components, and thus applied to the aircraft system. The proposed solutions showed interesting robustness features that were analysed by using a high–fidelity simulator, which was also able to include different operating points and realistic actuator faults, turbulence, measurement errors, and the model–reality mismatch. Future research directions will include the analysis of the proposed solutions when applied to real aircraft and spacecraft systems.

References

1. Babuška, R.: Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston (1998)
2. Baldi, P., Blanke, M., Castaldi, P., Mimmo, N., Simani, S.: Fault diagnosis for satellite sensors and actuators using nonlinear geometric approach and adaptive observers. Int. J. Robust Nonlinear **29**(16), 5429–5455 (2019). https://doi.org/10.1002/rnc.4083. Special Issue: Fault Diagnosis and Fault-Tolerant Control in Aerospace Systems
3. Baldi, P., Castaldi, P., Mimmo, N., Simani, S.: A new aerodynamic decoupled frequential FDIR methodology for satellite actuator faults. Int. J. Adapt. Control **28**(9), 812–832 (2014). https://doi.org/10.1002/acs.2379. Invited Paper for the Special Issue on "Emerging Trends in Active Methods for Fault Tolerant Control". John Wiley & Sons, Ltd. ISSN: 0890-6327
4. Benini, M., Castaldi, P., Simani, S.: Fault Diagnosis for Aircraft System Models: An Introduction from Fault Detection to Fault Tolerance, 1st edn. VDM Verlag Dr. Muller Aktiengesellschaft & Co. KG, Saarbrücken (2009). ISBN 978-3-639-21364-5. http://www.vdm-publishing.com/

5. Marcello, B., Paolo, C., Walter, G., Silvio, S.: Fault detection and isolation for on board sensors of a general aviation aircraft. Int. J. Adapt. Control **20**(8), 381–408 (2006). https://doi.org/10.1002/acs.906. Copyright 2006 John Wiley & Sons, Ltd. ISSN 0890-6327
6. Castaldi, P., Geri, W., Bonfe, M., Simani, S., Benini, M.: Design of residual generators and adaptive filters for the FDI of aircraft model sensors. Control Eng. Pract. **18**(5), 449–459 (2010). https://doi.org/10.1016/j.conengprac.2008.11.006. ACA'07 – 17th IFAC Symposium on Automatic Control in Aerospace Special Issue. Publisher: Elsevier Science. ISSN 0967-0661
7. Castaldi, P., Mimmo, N., Simani, S.: Differential geometry based active fault tolerant control for aircraft. Control Eng. Pract. **32**, 227–235 (2014). https://doi.org/10.1016/j.conengprac.2013.12.011
8. Castaldi, P., Mimmo, N., Simani, S.: Avionic air data sensors fault detection and isolation by means of singular perturbation and geometric approach. Sensors **17**(10), 2202 (2017). https://doi.org/10.3390/s17102202. Invited paper for the special issue "Models, Systems and Applications for Sensors in Cyber Physical Systems"
9. Fantuzzi, C., Simani, S., Beghelli, S., Rovatti, R.: Identification of piecewise affine models in noisy environment. Int. J. Control **75**(18), 1472–1485 (2002). https://doi.org/10.1109/87.865858
10. Simani, S., Castaldi, P.: Concepts and methods in fault tolerant control with application to a wind turbine simulated system. In: A Closer Look at Fault–Tolerant Control. Systems Engineering Methods, Developments and Technology, pp. 1–30, 1st edn. Nova Science Publishers, Hauppauge, June 2020. ISBN 978-1-53617-528-8
11. Simani, S., Fantuzzi, C., Rovatti, R., Beghelli, S.: Parameter identification for piecewise linear fuzzy models in noisy environment. Int. J. Approx. Reason. **1**(22), 149–167 (1999)
12. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Ann. Rev. Control **32**, 229–252 (2008)

Prediction of Discharge Capacity of Labyrinth Weir with Gene Expression Programming

Hossein Bonakdari[1(✉)], Isa Ebtehaj[1], Bahram Gharabaghi[2], Ali Sharifi[3], and Amir Mosavi[4,5,6]

[1] Department of Soils and Agri-Food Engineering, Laval University, Québec G1V0A6, Canada
hossein.bonakdari@fsaa.ulaval.ca
[2] School of Engineering, University of Guelph, Guelph, ON NIG 2W1, Canada
[3] Department of Statistics, Razi University, Kermanshah, Iran
[4] Department of Mathematics and Informatics, J. Selye University, 94501 Komarno, Slovakia
[5] Kalman Kando Faculty of Electrical Engineering, Obuda University, Budapest 1034, Hungary
[6] Institute of Structural Mechanics, Bauhaus-Universität Weimar, 99423 Weimar, Germany

Abstract. This paper proposes a model based on gene expression programming for predicting discharge coefficient of triangular labyrinth weirs. The parameters influencing discharge coefficient prediction were first examined and presented as crest height ratio to the head over the crest of the weir (p/y), crest length of water to channel width (L/W), crest length of water to the head over the crest of the weir (L/y), Froude number ($F = V/\sqrt{(gy)}$) and vertex angle (θ) dimensionless parameters. Different models were then presented using sensitivity analysis in order to examine each of the dimensionless parameters presented in this study. In addition, an equation was presented through the use of nonlinear regression (NLR) for the purpose of comparison with Gene Expression Programming (GEP). The results of the studies conducted by using different statistical indexes indicated that GEP is more capable than NLR. This is to the extent that GEP predicts discharge coefficient with an average relative error of approximately 2.5% in such manner that the predicted values have less than 5% relative error in the worst model.

Keywords: Discharge coefficient · Soft computing · Weir · Sensitivity analysis · Nonlinear regression

1 Introduction

Conventional weirs are structures used to control, regulate and measure water level and flow volume in irrigation and drainage networks and water and wastewater treatment plants. A conventional weir is usually installed along the flow and perpendicular to channel axis. Conventional weirs include rectangular, V-notch, labyrinth and complex weirs. Many theoretical and experimental studies investigated passing flow from conventional weirs. Taylor [1] presented an experimental study on hydraulic labyrinth weirs. Hay and Taylor [2] described how the head on the labyrinth weir effects the discharge ratio. Tullis et al. [3] investigated trapezoid labyrinth weirs and indicated that their discharge

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 202–217, 2021.
https://doi.org/10.1007/978-3-030-55180-3_17

capacity was a function of total head, effective length of weir crest and coefficient of discharge of labyrinth weir. Wormleaton and Soufiani [4] studied hydraulic features and aeration of triangle labyrinth weirs. They found that aeration efficiency of triangle labyrinth weirs is more than linear weirs with equal length. Also, Wormleaton and Tsang [5] studied aeration of rectangular weirs experimentally. Emiroglu and Baylar [6] investigated the effects of weir included angle and water sill slope of weir on aeration in triangle labyrinth weirs. Tullis et al. [7] studied hydraulic behavior and flow head on submerged labyrinth weirs. They concluded that the flow over submerged labyrinth weirs did not depend on labyrinth weir sidewall angles. Bagheri and Heidarpour [8] used free vortex theory to estimate discharge coefficient of sharp-crested rectangular weirs as a function of flow features, channel geometry and conventional weir. Kumar et al. [9] experimentally investigated discharging capacity of triangle labyrinth weirs. They suggested a relation to calculate the flow over triangle labyrinth weirs through analyzing experimental data.

Considering the complexity of engineering problems and the growing number of engineering studies, new methods called soft computing, were significantly used during recent decade that were more efficient and more accurate in solving complicated and difficult engineering issues and, facilitating studies [10–13]. Soft computing and artificial intelligence were used by different researchers to estimate and predict different hydraulic and hydrologic problems especially discharge coefficient [14–17]. Emiroglu et al. [18] used Adaptive Neuro Fuzzy Inference System (ANFIS) techniques to predict discharge capacity of the triangular labyrinth side weir. They introduced an equation for discharge coefficient in this type of side weirs. The diversion flow passing over sharp-crested rectangular side weirs were predicted using Feed Forward Neural Networks (FFNN) and Radial Basis Neural Networks (RBNN) by [19]. Bilhan et al. [19] introduced an equation for discharge coefficient as a function of geometric and hydraulic features for sharp-crested rectangular side weirs. Emiroglu et al. [20] used artificial neural networks to introduce a relation which calculated discharge coefficient of triangle labyrinth weirs located in rectangular in under critical flow conditions.

Gene Expression Programming (GEP) is one method used in water hydraulic engineering during recent years. Unlike artificial neural system and neuro fuzzy systems which include a black box, the suggested method showed high accuracy in estimating the given parameter and relation [21–25].

Using Gene Expression Programming (GEP), the present study aims to introduce an equation to predict discharge coefficient. Therefore, the parameters influencing discharge coefficient are first determined and then an equation is presented using GEP. Following that, the effect of each of the dimensionless parameters is examined on predicting discharge coefficient through using sensitivity analysis. Also, the results of the GEP model are compared with that of nonlinear regression (NLR). This paper is organized as follows: Sect. 2 presents data collection, Sect. 3 reviews GEP method, Sect. 4 comprises the discharge coefficient derivation based on GEP, Sects. 5 and 6 present the obtained results and conclusion, respectively.

2 Data Collection

The present study used Kumar et al. [9] experimental data to estimate the coefficient of discharge. A horizontal rectangular channel with 12 m length, 0.28 m width and 0.41 m depth was used in their tests. The used triangle weir was located 11 m away from the channel entrance. Water was provided for the channel through an inlet pipe from an overhead tank supplied with an overflow arrangement to keep a constant head. The water height over weir crest was measured by point gages having ±0.1 mm accuracy. Ventilation holes were installed on both sides of the weir's downstream for the purpose of aeration of the nappe. Wave suppressors and Grid walls were structured at the upstream of the channel to break and dissipate the surface disturbances and to enlarge the size of eddies, respectively. They conducted their experiments on 30, 60, 90, 120, 150, and 180° weirs. They also used varied discharges for each of the mentioned angles. They eventually carried out 123 different experiments for different discharges and angles. Schematic of Kumar et al. [9] experimental model is illustrated in Fig. 1. Table 1 shows the parameters used in the present study.

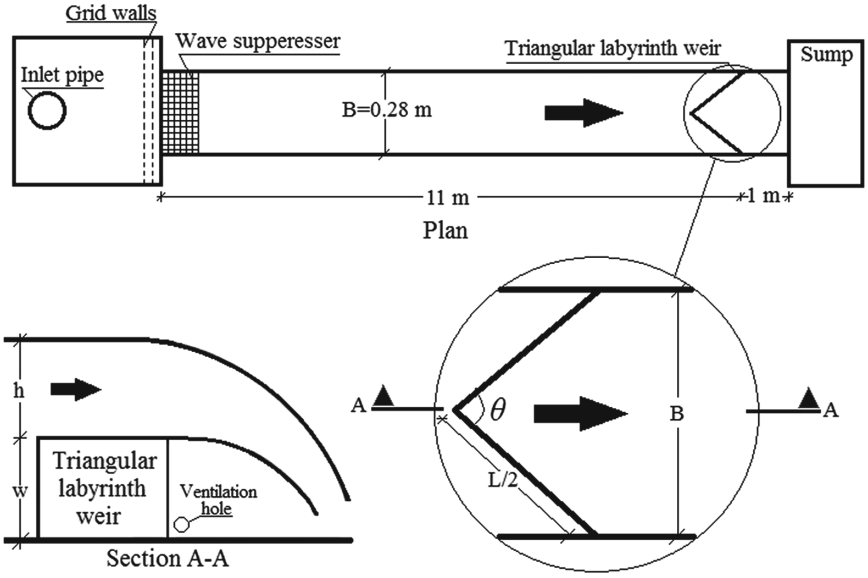


Fig. 1. Schematic of Kumar et al. [9] experimental model

Table 1. Parameters used to estimate discharge coefficient [9]

	p/y	L/W	F	W/y	θ (degree)	C_d
min	0.581	1	0.608	1.62	30	0.54
max	0.92	3.864	3.261	10.82	180	0.906

3 Overview of Gene Expression Programming

GEP is a developed genetic programming (GP) [26]. It is a search technique relying on computer programs such as decision tree, logical expressions, polynomial construct, and mathematics statements. GEP computer programs are coded as line chromosomes and the final presentation is in the form of expression trees (ETs) [27]. ETs are complex computer programs which are developed to solve a given problem and are selected according to their fitness to the problem [25]. Considering that in GP, genotype and phenotype are mixed in a simple replicator system, GEP of a genotype/phonotype system is developed where genotype is completely separated from phenotype. Therefore, developed GEP genotype/phonotype system is 100 to 60000 times more effective than GP system [28, 29].

In GEP process, the first chromosome of each independent parameter is randomly generated in the population. Then, they are developed and all independent parameters are evaluated based on fitness function and are used as a part to produce new generation with different characteristics. People of the new generation develop through confrontation with the selection environment, expression of the genomes and reproduction with modification. The process continues until getting the predefined generation or getting the answer [28, 29].

Ferreira [30] described the fitness of an individual function (i) for the fitness model

$$If \quad E(ij) \leq p, \; then\, f_{(ij)} = 1, \quad else \quad f_{(ij)} = 0 \tag{1}$$

where p and E(ij) are the precision and error, respectively. Then the absolute error can be obtained from:

$$E(ij) = \left| p_{(ij)} - T_j \right| \tag{2}$$

Where the (f_i) for an individual function calculated as follows:

$$f_i = \sum (R - |p_{(ij)} - T_j) \tag{3}$$

where T_j, R and $p_{(ij)}$ are the target values, selection range, and predicted values, respectively. Accordingly, the terminal set (T) and function set (F) are calculated to select the chromosomes. Figure 2 presents the GEP flowchart.

4 Derivation Discharge Coefficient Based on GEP

Reviewing the recent studies conducted on estimating discharge coefficient in weirs, crest height (p), head over the crest of the weir (y), crest length of the weir (L), channel width (W), and Froude number ($F = V/\sqrt{(gy)}$) parameters can be named [18–20, 31]. The dimensionless parameters in estimating discharge coefficient can be presented as Eq. (4) through using dimensional analysis.

$$C_d = f\left(\frac{w}{y}, \frac{L}{b}, \frac{L}{y}, F, \theta\right) \tag{4}$$

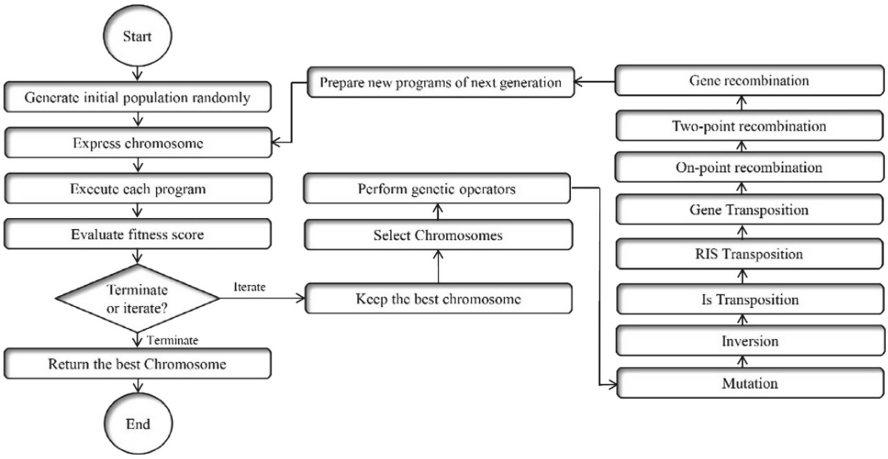


Fig. 2. Gene expression programming flowchart

The manner of function estimation through using the GEP method to predict discharge coefficient will be presented in this section. For testing, 20% of data set is used randomly as suggested by Kumar et al. [9]. Furthermore, 80% of data can be used for training. To produce an initial population of, according to Ferreira's [28] the range of 30–100 is suggested In the next step a fitness function is calculated using MSE as follows:

$$f_i = \frac{100}{1 + E_i} \quad for \quad E_i = p_{ij} - O_j \tag{5}$$

where P_{ij}, and Q_{ij} represent the predicted and fitness case values for i individual chromosome for fitness case j. The set of terminals are developed as follows:

$$T = \left\{ C_d, \frac{w}{y}, \frac{L}{b}, \frac{L}{y}, F, \theta \right\} \tag{6}$$

where the number of genes and their head and tail length are calculated for every chromosome. In the present study, three genes were used in each chromosome. In this study, the {+} operator is utilized to link function among the genes. The {x} function presented in Table 2 provides the $(1 - x)$ amount. Using Eq. (4) and the expression tree presented in Fig. 3, the model presented by using GEP can be expressed as Eq. (7); its parameters' values are presented in Table 3.

$$\begin{aligned} C_d = {} & Exp\left[F - \frac{L}{b} + 1.8\right] - Exp\left[1 - Exp\left[\frac{w}{y}\right]\right] \\ & + \frac{w}{y} \times Exp\left[0.034\frac{L}{y}(\theta - 1)\right] \\ & + 1 - \left[\frac{w}{y} + Exp\left[\frac{L}{b} + 1.58F - \theta + 1.79\right]\right] \end{aligned} \tag{7}$$

where C_d is coefficient of discharge, w/y the ratio of crest height to head over the crest of the weir, L/W ratio of crest length of water to channel width, L/y the ratio of

crest length of water to the head over the crest of the weir, F, Froude number and θ vortex angle.

Table 2. Parameters of GEP model

Parameter	Setting
Population size	50
Number of generations	40000
Number of chromosomes	40
Number of genes	3
Head size	4
function set	$\times$, $-$, $+$, Not, Exp, Pow
Linking function	Addition
Mutation rate	0.0014
Inversion rate	0.05
IS transposition rate	0.15
RIS transposition rate	0.15
Gene transposition rate	0.20
One-point recombination rate	0.15
Two-point recombination rate	0.15
Gene recombination rate	0.30

5 Result and Discussion

The accuracy of the model presented through the use of GEP (Eq. (7)) is examined in this section with using different statistical indexes. In addition, sensitivity analysis is also conducted in order to study the effect of each of the dimensionless parameter presented in predicting discharge coefficient. Following that the results from this model will also be compared with the results of the nonlinear regression analysis (NLR) to examine the accuracy of the model presented by using GEP.

In order to verify the accuracy of the estimated model at each step of model development, the results of analysis of GEP and NLR is based on the criteria of the coefficient of determination (R^2), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Adjusted Coefficient of Efficiency (CE) and Scatter Index (SI) as defined in the following forms:

$$R^2 = \left[\frac{\sum_{i=1}^{n} \left(C_{dEXP_i} - \overline{C_{dEXP}}\right)\left(C_{dGEP_i} - \overline{C_{dGEP}}\right)}{\sqrt{\sum_{i=1}^{n} \left(C_{dEXP_i} - \overline{C_{dEXP}}\right)^2 \sum_{i=1}^{n} \left(C_{dGEP_i} - \overline{C_{dGEP}}\right)^2}} \right]^2 \tag{8}$$

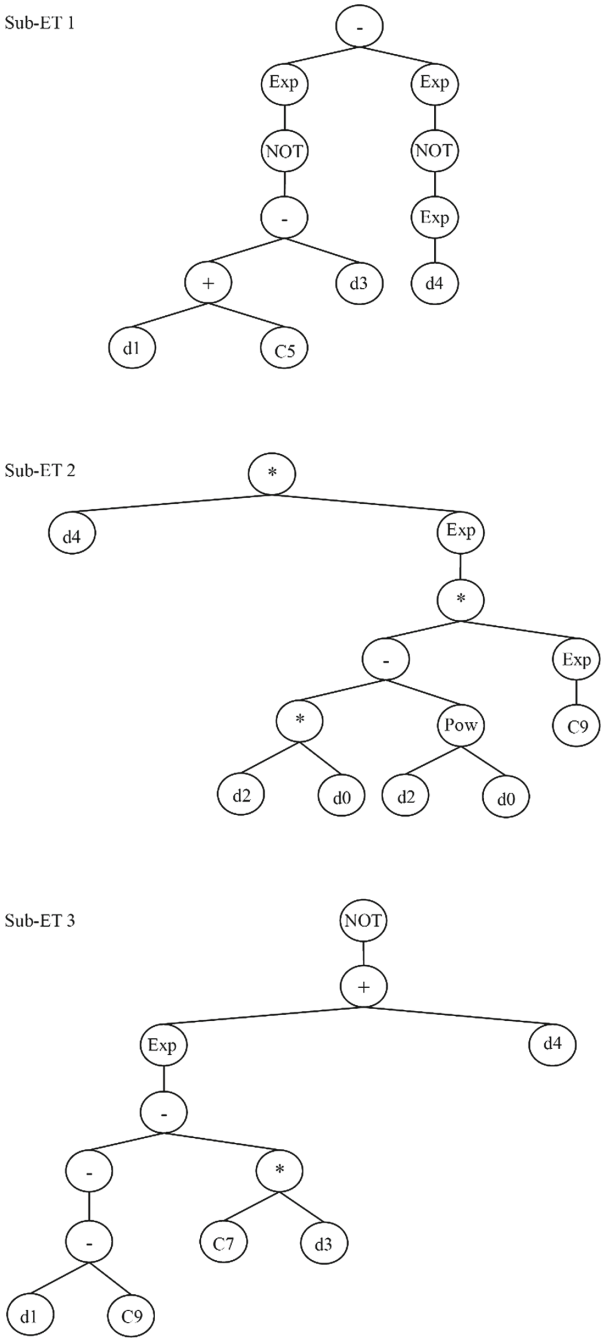


Fig. 3. Expression tree (ET) for presented model (Eq. (7))

Table 3. The values of the parameters used in ET (Fig. 3)

Parameter	Value	Parameter	Value
d0	θ	G1C5	2.8
d1	L/W	G2C9	−3.38
d2	L/y	G3C7	1.58
d3	F	G3C9	1.79
d4	p/y	–	–

$$RMSE = \sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(C_{dEXP_i} - C_{dGEP_i}\right)^2} \tag{9}$$

$$MAPE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(\frac{\left|C_{dEXP_i} - C_{dGEP_i}\right|}{C_{dEXP_i}}\right) \times 100 \tag{10}$$

$$CE = 1 - \frac{\sum_{i=1}^{n}\left|C_{dEXP_i} - C_{dEXP_i}\right|}{\sum_{i=1}^{n}\left|C_{dEXP_i} - \overline{C_{dEXP}}\right|} \tag{11}$$

$$SI = \frac{RMSE}{\overline{C_{dEXP}}} \tag{12}$$

where C_{dEXP_i} and C_{dGEP_i} denote the actual and modeled discharge coefficient values and $\overline{C_{dEXP}}$ and $\overline{C_{dGEP}}$ represent the mean actual and modeled discharge coefficient values, respectively.

The closer the value of index R^2 to 1, the more it shows the compatibility of the estimated value with the real value. Results which are obtained from coefficient of determination (R^2) have been simulated in relation with linear dependence between real and corresponding values (for the present case, the actual and simulated discharge coefficient values) and they are sensitive towards deviated points; so in evaluating the results, we cannot solely rely on this index. Thus, other statistical indexes like mean absolute percentage error (MAPE) - which shows the difference between real and estimated models in form of percentage of actual values- and root mean square error (RMSE) - which considers the weight of larger errors by powering the difference between actual and estimated values - are needed in order to estimate the function of the models. Both MAPE and RMSE indexes can include zero value (best mode) and infinity (worst value). Also, dimensionless RMSE criterion which is stated in SI form can be applied in estimating different models without considering dimension of parameters. Besides, as a complementary criterion, the "adjusted coefficient efficiency (CE)" could be utilized for evaluating the precision of models. This index reports the difference between the proportion of remainders variance (numerator term) and the data variance (denominator term) from 1. If this index equals 1, the presented model has done data estimation in the best way. Simultaneous use of these indexes could provide sufficient information for precision of the applied models [31, 32].

As mentioned earlier, the data utilized in this study is divided into two groups of "train" and "test" in such way that 20% of the data is selected through random selection without replacement for the purpose of testing, and the discharge coefficient parameter was presented as Eq. (7) using the remaining 80% data. Figure 4 shows the results obtained from training the presented GEP model in test and train states. The x axis indicates the actual values and y axis presents the values predicted by GEP. It could be seen in the figure that almost the majority of the predicted amounts predict the discharge coefficient fairly accurately in both states of test and train. The GEP model presented in the train predicts the train-state values with $R^2 = 0.95$ and an average relative error percentage approximate to 2% (MAPE). Most of the values presented in this state have a less than 5% relative error. The other statistical indexes used in the train state of this research are RMSE = 0.017, CE = 0.78 and SI = 0.02 indexes MAPE and RMSE have very low amounts - as can be seen almost zero - which indicates the high accuracy of the presented model. The predicted values have an $R^2 = 0.93$ and a MAPE = 2.53% in the test state which are almost similar to that of the train state. Also SI, CE, RMSE indexes are equal to 0.021, 0.67 and 0.029, respectively for the test state of this model. Therefore, considering Fig. 3 and the presented statistical indexes for train and test states of the presented GEP model, it could be stated that GEP predicts the discharge coefficient of triangular labyrinth weirs very well.

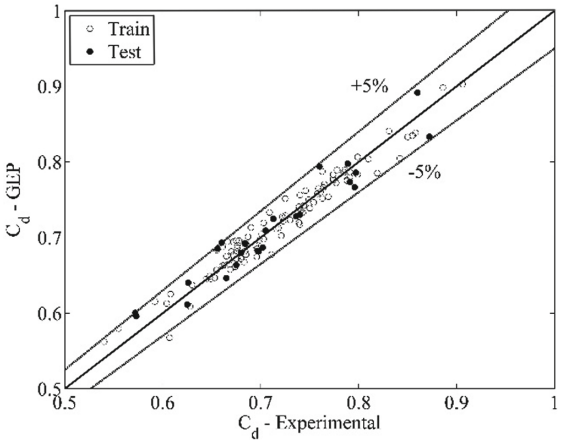


Fig. 4. Comparing estimated discharge coefficient with experimental result (test and train)

Through the use of sensitivity analysis in this section, the effect of each of the presented parameters is examined on predicting discharge coefficient of triangular labyrinth weirs. Therefore, different models are presented as Table 4. To estimate discharge in each of these models, the data is divided into two 80% and 20% groups, like they were in Eq. (7), for the purpose of training and testing the model, respectively. Tables 5 and 6 present the results of different statistical indexes, presented in the study, for the two "train" and "test" states, respectively. They demonstrate that the results of all the statistical indexes are better for model 1 when compared to the rest of the models for both train and test states. Also, Fig. 5 indicates that the maximum relative error of model 1

is lesser than all other models. Therefore, it could be stated that the simultaneous use of dimensionless parameters of crest height ratio to the head over the crest of the weir (p/y), crest length of water to channel width (L/W), crest length of water to the head over the crest of the weir (L/y), Froude number ($F = V/\sqrt{(gy)}$) and vertex angle (θ) is fixed in predicting discharge coefficient of rectangular labyrinth weirs. To examine the effect of each of the dimensionless parameters, the results of the statistical indexes of each model must be compared with regard to model 1 which is the best model and is presented as Eq. (7). It could be observed that model 2, which considers all the parameters of model 1 except for the vertex angle (θ), presents better results in comparison with models 3, 4, 5, and 6. Therefore, it could be stated that among the five presented dimensionless parameters, vertex angle (θ) parameter has the least value of effect on predicting discharge coefficient of triangular labyrinth weirs. Models 3, 4, 5 and 6 which disregard Froude number ($F = V/\sqrt{(gy)}$), crest length of water to the head over the crest of weir (L/Y), crest length of water to channel width (L/w), and crest height ratio to the head over the crest (p/y) dimensionless parameters respectively, do not present better results in comparison with models 1 and 2. Therefore, not using these parameters prevents predicting discharge coefficient relatively accurately in such manner that in some cases their maximum relative error is approximately 20% regarding Fig. 5. Therefore, it is essential to use these parameters in predicting discharge coefficient.

Table 4. Dependent parameters in discharge coefficient prediction

Independent parameter	Dependent parameter	Model no.
p/y, L/W, L/y, F, θ	C_d	1
p/y, L/W, L/y, F	C_d	2
p/y, L/W, L/y, θ	C_d	3
p/y, L/W, F, θ	C_d	4
p/y, L/y, F, θ	C_d	5
L/W, L/y, F, θ	C_d	6

Table 5. Statistics indexes (train)

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
R^2	0.95	0.91	0.68	0.7	0.84	0.68
RMSE	0.017	0.021	0.055	0.040	0.028	0.039
MAPE (%)	1.920	2.442	6.139	4.379	2.823	4.452
CE	0.780	0.663	0.314	0.480	0.640	0.234
SI	0.020	0.029	0.076	0.055	0.039	0.054

Table 6. Statistics indexes (test)

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
R^2	0.93	0.88	0.73	0.76	0.88	0.63
RMSE	0.021	0.026	0.054	0.040	0.028	0.047
MAPE (%)	2.538	3.004	6.142	4.891	3.056	5.327
CE	0.699	0.652	0.375	0.505	0.665	0.202
SI	0.029	0.037	0.076	0.055	0.039	0.065

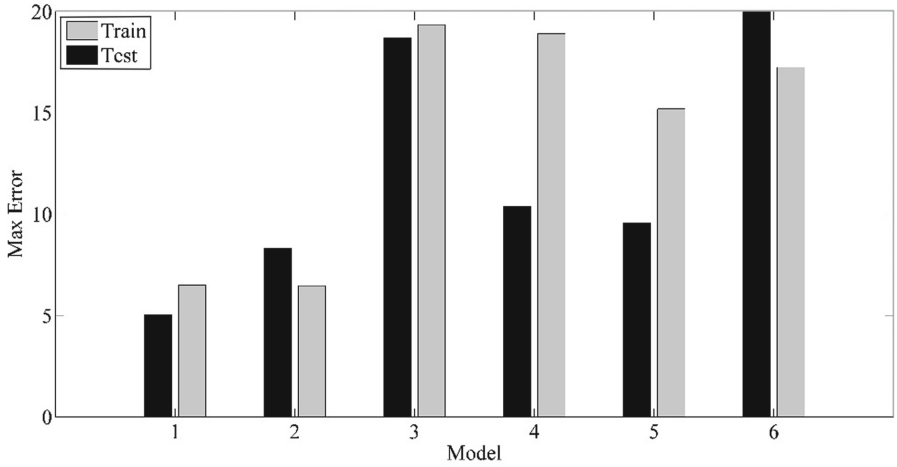


Fig. 5. Highest errors in six different models

Also, this study presents an equation (Eq. (13)) that employs nonlinear regression (NLR) in MINITAB to predict discharge coefficient of triangular labyrinth. The set of data selected to train GEP were also used in this state in predicting the following equation. Also, through employing the data used by random selection without replacement for testing GEP, the accuracy of the following equation is used in this section.

$$C_d = 0.466 + 0.338\,(p/y) - 0.183\,(L/W) - 0.022\,(L/y) + 0.31F + 0.12sin(\theta) \tag{13}$$

Figure 6 shows the results of discharge coefficient prediction for the two presented models using GEP and NLR. The x axis of this figure shows the experimental values (Target) and the y axis shows the values predicted through using GEP and NLR methods. The data used in this figure had no role in estimating Eq. (7) and (13) and as mentioned in the previous sections they were selected using random selection without replacement for the purpose of testing the model. The figure indicates that the equation presented by using GEP (Eq. (7)) is fairly accurate in predicting discharge coefficient in a way that it predicts all the predicted discharge coefficients with a relative error less than 5%. This figure also shows that the equation presented by using NLR mostly presents the discharge coefficient to be less than the actual value which leads to underestimating

the prediction of the passing discharge and so causes underestimating. It could also be observed that the predicted values have a relative error greater than 5% in this state as opposed to GEP equation.

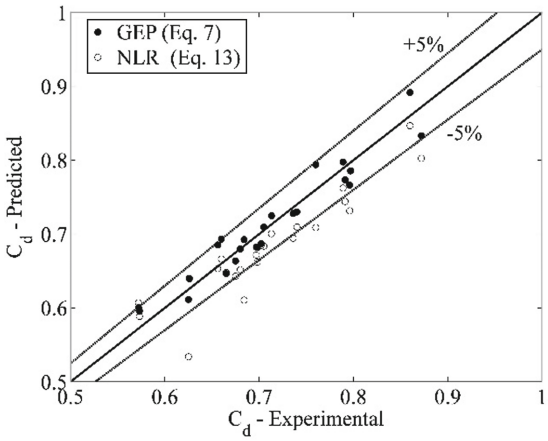


Fig. 6. Comparison of GEP and NLR in prediction of discharge coefficient of triangular labyrinth weirs (test)

Table 7 shows the results of the statistical indexes presented in this study in order to verify the accuracy of the equations presented by using GEP and NLR in predicting discharge coefficient for both states of train and test. Careful consideration of the table indicates that R^2 is more and less than 0.9 in both states of train and test of GEP and NLR respectively. It could also be seen that the average relative error is approximately 2.5% for GEP in test state and it is almost 4.5% for NLR. It is also observed that the results of RMSE and SI indexes for GEP are less than NLR and considering the fact that approaching these two indexes to zero indicates the higher accuracy of the model, it could be stated that the GEP model presented in this study is relatively less accurate with regard to the results obtained from NLR. The values predicted using Eqs. (7), (GEP), and (13), (NLR), are presented in Table 8 for different hydraulic conditions.

Table 7. Comparing different statistical indexes for the discharge coefficients predicted by using GEP and NLR

Statistics indexes	Train		Test	
	GEP (Eq. 7)	NLR (Eq. 13)	GEP (Eq. 7)	NLR (Eq. 13)
R^2	0.95	0.78	0.93	0.86
RMSE	0.015	0.044	0.021	0.040
MAPE (%)	1.620	4.664	2.538	4.583
CE	0.780	0.341	0.699	0.495
SI	0.020	0.061	0.029	0.055

Table 8. Predicted coefficient of discharge using GEP and NLR

θ (degree)	L (m)	w (m)	y (m)	Q (m^3/s)	C_d (Exp)	C_d (GEP)	C_d (NLR)
30	1.082	0.092	0.011	0.003	0.86	0.892	0.847
30	1.082	0.092	0.017	0.006	0.76	0.794	0.709
30	1.082	0.092	0.026	0.009	0.684	0.693	0.611
30	1.082	0.092	0.032	0.012	0.625	0.611	0.534
60	0.56	0.101	0.013	0.002	0.872	0.833	0.803
60	0.56	0.101	0.031	0.006	0.705	0.709	0.684
60	0.56	0.101	0.051	0.011	0.573	0.596	0.588
60	0.56	0.101	0.029	0.006	0.713	0.725	0.701
90	0.396	0.103	0.014	0.002	0.789	0.798	0.762
90	0.396	0.103	0.047	0.008	0.702	0.687	0.685
90	0.396	0.103	0.069	0.012	0.572	0.6	0.607
90	0.396	0.103	0.058	0.01	0.626	0.64	0.639
120	0.323	0.106	0.027	0.003	0.791	0.773	0.744
120	0.323	0.106	0.044	0.007	0.74	0.73	0.710
120	0.323	0.106	0.073	0.012	0.665	0.646	0.648
120	0.323	0.106	0.06	0.01	0.697	0.682	0.672
150	0.29	0.108	0.014	0.001	0.797	0.786	0.785
150	0.29	0.108	0.071	0.011	0.698	0.682	0.662
150	0.29	0.108	0.034	0.004	0.796	0.766	0.731
150	0.29	0.108	0.052	0.008	0.736	0.728	0.694
180	0.28	0.1	0.055	0.007	0.656	0.685	0.653
180	0.28	0.1	0.072	0.011	0.675	0.664	0.643
180	0.28	0.1	0.045	0.005	0.66	0.693	0.666
180	0.28	0.1	0.061	0.008	0.68	0.68	0.652

Considering the estimation of coefficient of discharge relation and discharge equation on sharp-crested weir under free flow in channel, defined as follow, Eq. (7) shows the outflow as:

$$Q = \frac{2}{3} C_d \sqrt{2g} L y^{1.5} \tag{14}$$

where C_d is coefficient of discharge, w/y the ratio of crest height to head over the crest of the weir, L/W ratio of crest length of water to channel width, L/y the ratio of crest length of water to the head over the crest of the weir, F Froude number, L crest length of water, y head over the crest of the weir, g acceleration due to gravity and θ vertex angle.

6 Conclusions

There are many ways to control flood such as using weirs which are either located aside or along the channel. To predict the coefficient of discharge of a weir along the channel, the present study made use of the ratio of crest height to head over the crest of the weir (p/y), crest length of water to channel width (L/W), crest length of water to the head over the crest of the weir (L/y), Froude number ($F = V/\sqrt{(gy)}$) and vortex angle (θ) and an equation has been presented as Eq. (7) using GEP. The accuracy of the presented model was examined through taking different statistical indexes into consideration and the results indicated that Eq. (7) predicts discharge coefficient with an approximate relative error of 2.5% for hydraulic conditions which had no role in training the model. Also, the amounts of all the C_d predicted through using this method had a relative error less than 5%. Following that, different models were presented in order to examine the effect of each of the dimensionless parameters presented in this study. The results demonstrate that vortex angle (θ) parameter had lesser effect in predicting C_d in comparison with the other models. Also, the simultaneous use of crest height ratio to the head over the crest of the weir (p/y), crest length of water to channel width (L/W), crest length of water to head over the crest of weir (L/W), Froude number ($F = V/\sqrt{(gy)}$), and vortex angle (θ) dimensionless parameters is necessary in predicting the discharge coefficient. Then, in order to examine the accuracy of the models presented by using GEP, in comparison with nonlinear regression analysis (NLR), an equation was presented through using NLR as Eq. 13 and the results indicated the higher accuracy of GEP in comparison with NLR. For future works, it is recommended to apply other techniques such as multi expression programming and compared the results with the results of the developed model in the current study.

Acknowledgment. We acknowledge the financial support of this work by the Hungarian State and the European Union under the EFOP-3.6.1-16-2016-00010 project and the 2017-1.3.1-VKE-2017-00025 project. We also acknowledge the support of the German Research Foundation (DFG) and the Bauhaus-Universität Weimar within the Open-Access Publishing Programme.

References

1. Taylor, G.: The performance of labyrinth weir. Ph.D. thesis, University of Nottingham, Nottingham, England (1968)
2. Hay, N., Taylor, G.: A computer model for the determination of the performance of labyrinth weirs. In: 13th Congress of IAHR, Koyoto, Japan, pp. 361–378 (1969)
3. Tullis, J.P., Amanian, N., Waldron, D.: Design of labyrinth spillways. J. Hydraul. Eng. **121**(3), 247–255 (1995)
4. Wormleaton, P.R., Soufiani, E.: Aeration performance of triangular planform labyrinth weirs. J. Environ. Eng. **124**(8), 709–719 (1998)
5. Wormleaton, P.R., Tsang, C.C.: Aeration performance of rectangular planform labyrinth weirs. J. Environ. Eng. **126**(5), 456–465 (2000)
6. Emiroglu, M.E., Baylar, A.: Influence of included angle and sill slope on air entrainment of triangular planform labyrinth weirs. J. Hydraul. Eng. **131**(3), 184–189 (2005)

7. Tullis, B.P., Young, J.C., Chandler, M.A.: Head-discharge relationships for submerged labyrinth weirs. J. Hydraul. Eng. **133**(3), 248–254 (2007)
8. Bagheri, S., Heidarpour, M.: Application of free vortex theory to estimating discharge coefficient for sharp-crested weirs. Biosys. Eng. **105**(3), 423–427 (2010)
9. Kumar, S., Ahmad, Z., Mansoor, T.: A new approach to improve the discharging capacity of sharp-crested triangular plan form weirs. Flow Meas. Instrum. **22**(3), 175–180 (2011)
10. Ebtehaj, I., Bonakdari, H.: Bed load sediment transport estimation in a clean pipe using multilayer perceptron with different training algorithms. KSCE J. Civil Eng. **20**(2), 581–589 (2016). https://doi.org/10.1007/s12205-015-0630-7
11. Bonakdari, H., Ebtehaj, I.: Verification of equation for non-deposition sediment transport in flood water canals. In: 7th International conference on fluvial hydraulics, River Flow, pp. 1527–1533 (2014)
12. Azimi, H., Bonakdari, H., Ebtehaj, I., Talesh, S.H.A., Michelson, D.G., Jamali, A.: Evolutionary Pareto optimization of an ANFIS network for modeling scour at pile groups in clear water condition. Fuzzy Sets Syst. **319**, 50–69 (2017)
13. Ebtehaj, I., Bonakdari, H.: Evaluation of sediment transport in sewer using artificial neural network. Eng. Appl. Comput. Fluid Mech. **7**(3), 382–392 (2013)
14. Azimi, H., Bonakdari, H., Ebtehaj, I.: Sensitivity analysis of the factors affecting the discharge capacity of side weirs in trapezoidal channels using extreme learning machines. Flow Meas. Instrum. **54**, 216–223 (2017)
15. Azimi, H., Bonakdari, H., Ebtehaj, I.: Design of radial basis function-based support vector regression in predicting the discharge coefficient of a side weir in a trapezoidal channel. Appl. Water Sci. **9**(4), 1–12 (2019). https://doi.org/10.1007/s13201-019-0961-5
16. Azimi, H., Shabanlou, S., Ebtehaj, I., Bonakdari, H., Kardar, S.: Combination of computational fluid dynamics, adaptive neuro-fuzzy inference system, and genetic algorithm for predicting discharge coefficient of rectangular side orifices. J. Irrig. Drain. Eng. **143**(7), 04017015 (2017)
17. Ebtehaj, I., Bonakdari, H., Gharabaghi, B.: Development of more accurate discharge coefficient prediction equations for rectangular side weirs using adaptive neuro-fuzzy inference system and generalized group method of data handling. Measurement **116**, 473–482 (2018)
18. Emiroglu, M.E., Kisi, O., Bilhan, O.: Predicting discharge capacity of triangular labyrinth side weir located on a straight channel by using an adaptive neuro-fuzzy technique. Adv. Eng. Softw. **41**(2), 154–160 (2010)
19. Bilhan, O., Emiroglu, M.E., Kisi, O.: Application of two different neural network techniques to lateral outflow over rectangular side weirs located on a straight channel. Adv. Eng. Softw. **41**(6), 831–837 (2010)
20. Emiroglu, M.E., Bilhan, O., Kisi, O.: Neural networks for estimation of discharge capacity of triangular labyrinth side-weir located on a straight channel. Expert Syst. Appl. **38**(1), 867–874 (2011)
21. Azimi, H., Bonakdari, H., Ebtehaj, I.: Gene expression programming-based approach for predicting the roller length of a hydraulic jump on a rough bed. ISH J. Hydraul. Eng., 1–11 (2019). https://doi.org/10.1080/09715010.2019.1579058
22. Bonakdari, H., Gharabaghi, B., Ebtehaj, I.: A highly efficient gene expression programming for velocity distribution at compound sewer channel. In: The 38th IAHR World Congress from September 1st to 6th, Panama City, Panama (2019). https://doi.org/10.3850/38WC092019-0221
23. Ebtehaj, I., Bonakdari, H.: No-deposition sediment transport in sewers using gene expression programming. J. Soft Comput. Civil Eng. **1**(1), 29–53 (2017)
24. Khozani, Z.S., Bonakdari, H., Ebtehaj, I.: An analysis of shear stress distribution in circular channels with sediment deposition based on gene expression programming. Int. J. Sedim. Res. **32**(4), 575–584 (2017)

25. Khozani, Z.S., Bonakdari, H., Ebtehaj, I.: An expert system for predicting shear stress distribution in circular open channels using gene expression programming. Water Sci. Eng. **11**(2), 167–176 (2018)
26. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. A Bradford Book, MIT Press, Cambridge (1992)
27. Azamathulla, H.M., Ahmad, Z., Ghani, A.A.: computation of discharge through side sluice gate using gene-expression programming. Irrig. Drain. **62**(1), 115–119 (2013)
28. Ferreira, C.: Gene expression programming in problem solving, invited tutorial of the 6th online world conference on soft computing in industrial applications. In: Origins of Functionalist Theory, vol. 9, pp. 10–24 (2001)
29. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst. **13**, 87–129 (2001)
30. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, 2nd edn. Springer, Germany (2006)
31. Dursun, O.F., Kaya, N., Firat, M.: Estimating discharge coefficient of semi-elliptical side weir using ANFIS. J. Hydrol. **426**, 55–62 (2012)
32. Legates, D.R., McCabe Jr., G.J.: Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resour. Res. **35**(1), 233–241 (1999)

Biologically Inspired Exoskeleton Arm Enhancement Comparing Fluidic McKibben Muscle Insertions for Lifting Operations

Ravinash Ramchender[1](✉) and Glen Bright[2]

[1] Howard College, UKZN, Durban, South Africa
ravinashramchender@gmail.com
[2] Howard College, UKZN, 238 Mazisi Kunene Rd., Glenwood, Durban, South Africa
Bright@ukzn.ac.za

Abstract. Exoskeletons are being used in the industry to help individuals improve their endurance while reducing their chance of injury. These exoskeletons are based on human anatomy. To obtain the workspace of an exoskeleton, mathematical models are developed and simulated which is then refined to get the desired movements of an exoskeleton. Some of these exoskeletons are mechanically operated while others use complex control systems to perform a selected task. This can be achieved by using a combination of mechanical designs and electronics. Research of how exoskeleton functions and the types of exoskeletons are presented to achieve an overall workable exoskeleton. By implementing human assistive exoskeletons in industry, the unemployment rate can be decreased while creating the symbiotic association between man and machine. This study was conducted to determine the functionality of biologically inspired exoskeletons in an industrial environment.

Keywords: Kinematics · McKibben muscle · Workspace analysis · Pneumatic artificial muscles · Conserved energy · Bio-mechatronics

1 Introduction

Research suggests that human capabilities are limited due to their physical body structure [3]. With the increased technological development in industry, jobs that were once done by people are now discarded since technology is cheaper to operate as compared to the cost of human labour in the long run. Machines today are capable of performing tasks more efficiently with excellent accuracy and precision. These are the qualities that are favourable for industrial progression. According to recent statistics, South Africa is currently experiencing its highest unemployment rate of 29% since 2003 [3]. South Africa cannot cope with the fourth industrial revolution. These results are due to the lack of skills of individuals as well as the country not being prepared for technological development [3]. Therefore, by technologically enhancing one's capability to perform a task more efficiently and effortlessly, the rise in human and machine association can further be improved with harmonious results. This association can lead to the first step in creating the symbiosis between man and machine.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 218–231, 2021.
https://doi.org/10.1007/978-3-030-55180-3_18

The main objectives are to determine the mechanical and biological operation of the arm, design an exoskeleton that simulates the motion of a biological arm through kinematic models, assess the lifting system and its complexities. Also to compare fluidic insertions for a Pneumatic artificial muscle.

The design of the exoskeleton was specified to provide safe lifting applications, portability, must be adjustable for a range of different users, have a functional operation and assist with lifting at desired periods. The overall design of an exoskeleton was determined by the limitations of the user such as the operating workspace and therefore required some restrictions such as a range of motion and restricted rotations to prevent injury during operation.

This report highlights the mathematical model which best describes the configuration of the exoskeleton, the model that describes the operation of the artificial muscle and the control system in which the biologically inspired exoskeleton will operate.

2 Exoskeletons

The exoskeleton was intended to assist with over the waist tasks by improving the performance of the arm and the lower back. This was done whilst using fluidic muscles to act as the actuating system. There are exoskeleton suits that are available however, these commercially available body enhancements suits are either too expensive or lack the power to support large weights or are rather too bulky [1]. Therefore, there was a need to develop an exoskeleton suit that was capable of assisting with several tasks and was also compact. It was therefore decided that a single exoskeleton arm prototype will be developed to achieve a highly suitable body enhancement for industrial use. The outcome of this design will result in a combination of biological and mechatronic systems, also known as a Bio-Mechatronic system as seen in Fig. 1. A Bio-Mechatronic system is the advancement of mechatronic systems through the development of biologically inspired designs [2].

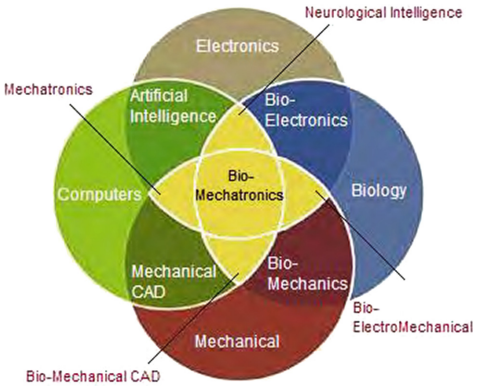


Fig. 1. Graphical representation of a bio-mechatronic system.

3 Wearable Exoskeletons

Exoskeletons are classified according to their functional purpose and how the exoskeleton operates. Some of these categories are body dependent parts that need to be actuated, static or dynamic operation, mobility and powered exoskeletons or mechanically operated exoskeletons [4]. These contribute to the type of exoskeleton that will be developed.

3.1 Body Fixtures

Some of the factors that needed to be considered as the movement and range of sizes of the body among different users. These are used to determine how many degrees of freedom is present in the arm and how much rotation is allowed as well as the body geometry relative to a fixed body dimension. The range of motion present in an individual's upper body can be seen in Fig. 2 [5].

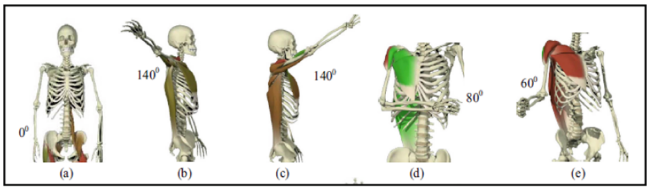


Fig. 2. Movement of upper limbs for everyday use.

3.2 Body Geometry

The average length of a person's body part can be determined by using the height of that individual. This is done by using the relationship between the height of an individual and their associated body parts [6]. This can be seen in Fig. 3.

3.3 Spinal Configuration

Weight distribution is essential when lifting loads that are too excessive for the human body. The spinal configuration demonstrates which vertebrae support the weight and the vertebrae that assist with upright posture. The vertebrae that assist with the weight distribution is the lumbar region, specifically the L4 and L5 (bottom two vertebrae of the lumbar region) vertebrae that transmit the weight towards the legs and feet as seen in Fig. 4. This was considered for the effective weight distribution of loads that are being carried towards the lower back to prevent any injury and strain of the user [7].

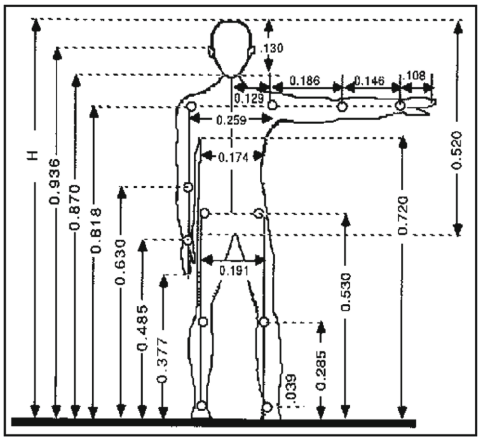


Fig. 3. Body parameters

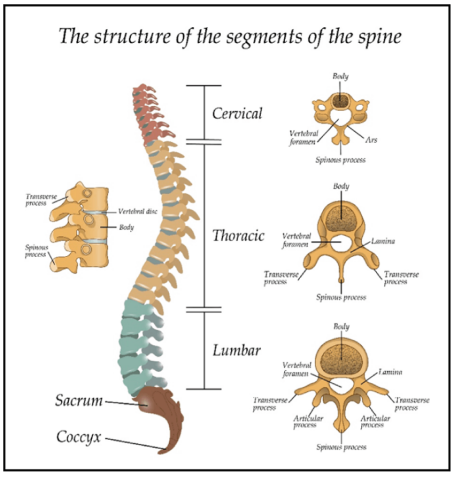


Fig. 4. Spinal configuration

3.4 Pneumatic Fluidic Muscle Based Exoskeleton Suit

A fluidic muscle-based exoskeleton arm was considered as it makes use of Festo muscles to operate the exoskeleton. These muscles are used in pairs to create an antagonistic setup which simulates the operation of the bicep and triceps of an individual. The exoskeleton also effectively distributes the weight carried by the user towards the lower limbs as seen in Fig. 5 [8].

3.5 Types of Pneumatic Muscle Actuators (PMA)

Several types of PMA's perform the same type of actuation, however, they vary in their geometry and their mechanical make-up. There are three types of muscles which are

Fig. 5. Festo based exoskeleton

commonly known today. These are the McKibben muscles, Netted muscles and the Festo embedded muscles as seen in Fig. 6 [9].

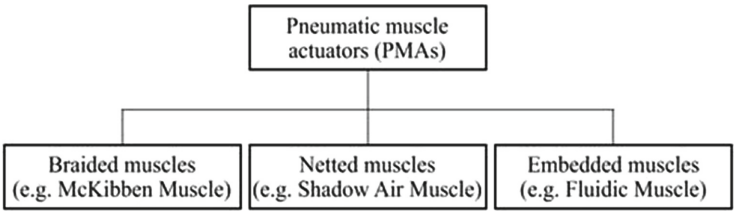


Fig. 6. Classification of PMA's

The type of PMA that was used was dependent on the workspace provided and the cost which was involved in manufacturing these muscles. Therefore, after consideration, a McKibben muscle was used.

4 Kinematic Models

A model of the exoskeleton which best fits the design requirement was modelled to mathematically determine its configuration and its operating space. The model which was used can be seen in Fig. 7. This model utilized 3 DOF to simulate the motion of an arm. The degrees of freedom associated with the arm is limited to prevent the rotation of the arm that cannot be controlled whilst carrying large loads.

4.1 D-H Parameters

The forward kinematics was resolved using the D-H method. This method comprises of four parameters which define the rotation and displacements of each link concerning their individual co-ordinate systems as seen in Fig. 7. The exoskeleton configuration resulted in the following D-H parameters as seen in Table 1:

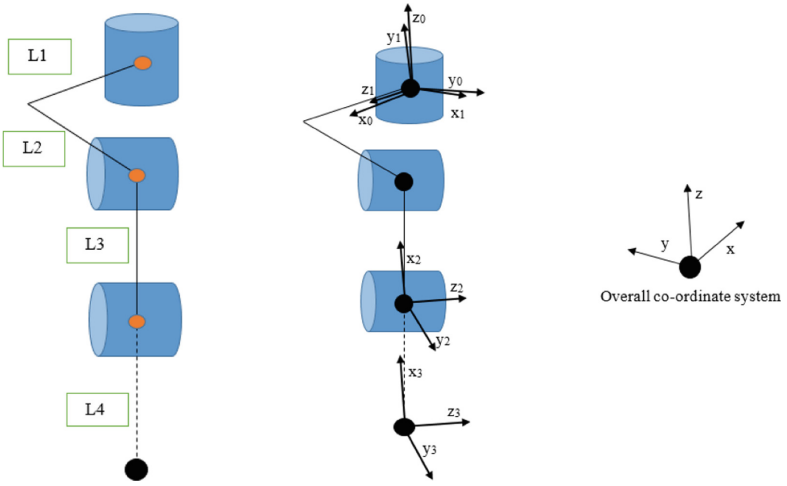


Fig. 7. Exoskeletal arm model

Table 1. D-H parameters

	θ	α	r	d
1	$-90° + \theta_1$	90°	L_2	0
2	$90° + \theta_2$	180°	L_3	L_1
3	θ_3	0	L_4	0

Where theta and alpha represent the rotation of the joints about a reference co-ordinate, and r and d are the displacements of the joints to the associated reference coordinate system. Using these D-H parameters, a homogeneous transformation matrix can be derived using the D-H Homogeneous matrix equation as seen in Eq. 1.

$$H_n^{n-1} = \begin{bmatrix} c(\theta_n) & -s(\theta_n)c(\alpha_n) & s(\theta_n)s(\alpha_n) & r_n c(\theta_n) \\ s(\theta_n) & c(\theta_n)c(\alpha_n) & -c(\theta_n)s(\alpha_n) & r_n s(\theta_n) \\ 0 & s(\alpha_n) & c(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The homogeneous equation that defines the 3 DOF model is as follows:

From 0–1

$$H_1^0 = \begin{bmatrix} c1 & 0 & s1 & L_2 c1 \\ s1 & 0 & -c1 & L_2 s1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

From 1–2

$$H_2^1 = \begin{bmatrix} c2 & s2 & 0 & L_3c2 \\ s2 & -c2 & 0 & L_3s2 \\ 0 & 0 & -1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

From 2–3

$$H_3^2 = \begin{bmatrix} c3 & -s2 & 0 & L_4c3 \\ s3 & c3 & 0 & L_4s3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The total homogeneous matrix that determines the rotation and co-ordinates of this model is:

$$H_3^0 = H_1^0 H_2^1 H_3^2 \tag{5}$$

4.2 Geometric Method

The inverse kinematics was resolved using a Geometric approach where trigonometric equations are used to define the geometry of the model. Looking at the model from the side view, the geometry required is formed and can be seen in Fig. 8.

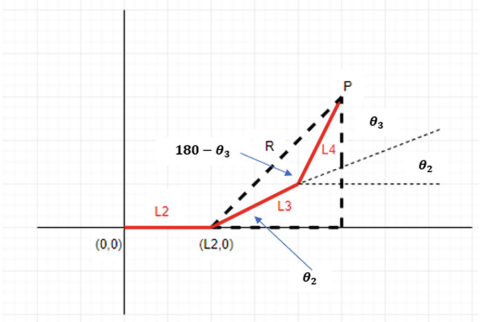


Fig. 8. Side view geometry of 3 DOF model.

Using the above geometry, the equations that form which relate the x and y co-ordinate to θ_2 and θ_3 are:

$$\boldsymbol{R^2 = L_3^2 + L_4^2 - 2L_3L_4\cos(180 - \theta_3)} \tag{6}$$

$$\boldsymbol{R^2 = (x - L_2)^2 + y^2} \tag{7}$$

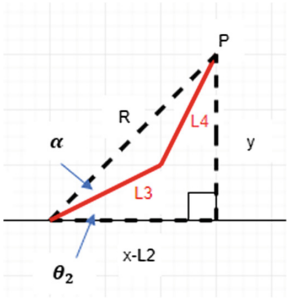


Fig. 9. Side view geometry of 3 DOF model.

Another triangle that is formed from the same view can be seen in Fig. 9. The equations obtained from this diagram is as follows:

$$L_4^2 = R^2 + L_3^2 - 2RL_3 \cos(\alpha) \tag{8}$$

$$\sin(\alpha + \theta_2) = \frac{y}{R} \tag{9}$$

From this θ_2 and θ_3 can be determined using only the link dimensions and associated end effector point.

4.3 Model Workspace

Using D-H Parameters, the workspace model for the exoskeleton arm was determined. This further showed the space required by the exoskeleton when it's allowed to rotate freely without any restrictions as seen in Fig. 10A. A valve of 1 unit was used to represent all link lengths to better visualize the exoskeleton. Looking at the top view of the workspace, the singularity of the model becomes apparent as there is a hole visible through the workspace of the model. This hole represents the space the exoskeleton arm occupies as seen in Fig. 10B.

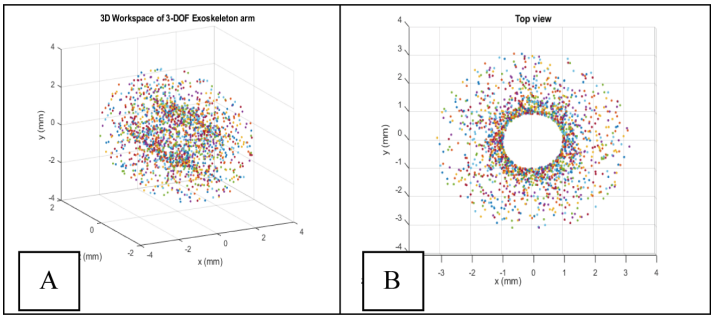


Fig. 10. Workspace model of exoskeleton without any constraints.

By apply constraints to limit the movement of the exoskeleton, the exoskeleton can function more efficiently without any collision with the user. The constraints applied are seen in Table 2:

Table 2. Constraints imposed.

L1	$0 < \theta_1 < 45$ (degrees)
L2	$-45 < \theta_2 < 180$ (degrees)
L3	$0 < \theta_3 < 150$ (degrees)

When considering the actual workspace of the model that was restricted to avoid any interference with the user, the workspace takes the shape of a crescent which can be seen in Fig. 11B. This allows the user to perform movements that are consistent with the human body.

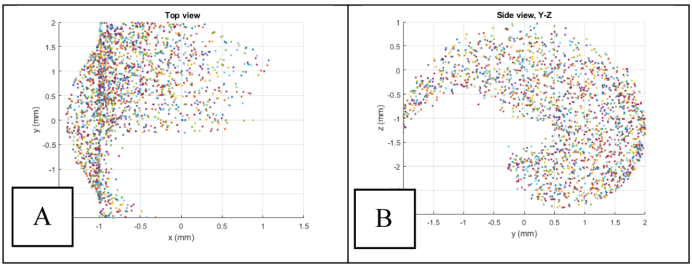


Fig. 11. Workspace of exoskeleton model with constraints.

5 McKibben Muscles

5.1 Static Modelling of McKibben Muscles

McKibben muscles function as linear actuators which converts pneumatic energy into mechanical energy. These muscles work on the principle of work in equals work out where energy is conserved. This forms an equation where a change in volume results in a force over a distance.

$$-FdL = PdV \tag{10}$$

However, by including an elastic force which is experienced by the elastic tube on the inside, we get:

$$F = -P\frac{dV}{dL} - \frac{dE}{dL} \tag{11}$$

This was further developed using the corresponding volume, length and elastic equations:

$$V = \frac{1}{4}\pi D^2 L = \frac{Ls^3}{4\pi N^2} \sin^2\theta \cos\theta \tag{12}$$

L_s being the strand length, N being the total encirclements of the strand and θ representing the braid angle.

$$L = L_s \cos\theta \tag{13}$$

$$E = \frac{1}{2}k(-\varepsilon L)^2 \tag{14}$$

The overall equation which represents the tension force of the muscle is as follows:

$$F = \frac{PD\pi}{4}\left(\frac{3}{\tan^2\theta} - \frac{1}{\sin^2\theta}\right) - \frac{DN\pi k}{\tan\theta}\varepsilon^2 \tag{15}$$

5.2 Fluidic Insertions

An initial test using air as the fluid insertion for a Festo muscle and a sleeve braided muscle was carried out. It can be seen that both the initial contraction and the release of the muscle tension demonstrates a graph which experiences hysteresis as seen in Fig. 12. Both muscles take longer to return to its original length after being pressurized. This is due to the elastic behaviour of the inner tubes.

Both muscles are of similar lengths and it is apparent that the contraction rate for both is consistent with an approximate contraction of 19%. Festo muscles are expensive and therefore Mckibben muscles are considered as they are easy and cheap to manufacture while providing similar contraction rates.

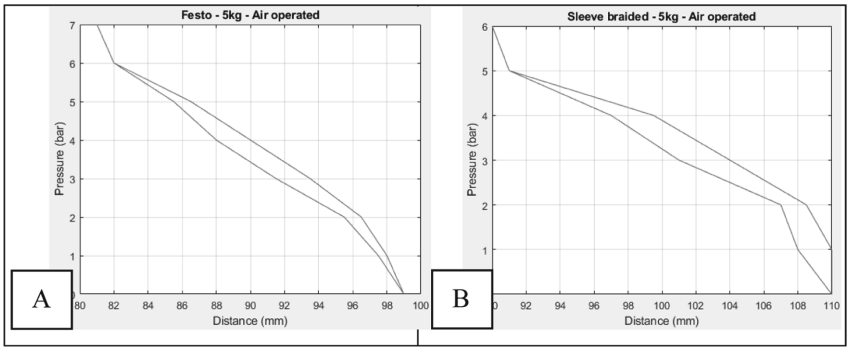


Fig. 12. Graphs showing hysteresis of muscles.

To optimize an adequate fluid which is used as the driving agent for the muscle, several fluids are tested. This was done to compare the contraction with that of the

pressurized air contraction. This is also done to see if a liquid insertion can be used to provide portability instead of using compressors to build up pressurized air. The other fluids used are a range of oils with varying viscosity. The graphs that depict the contraction of the muscle can be seen in Fig. 13.

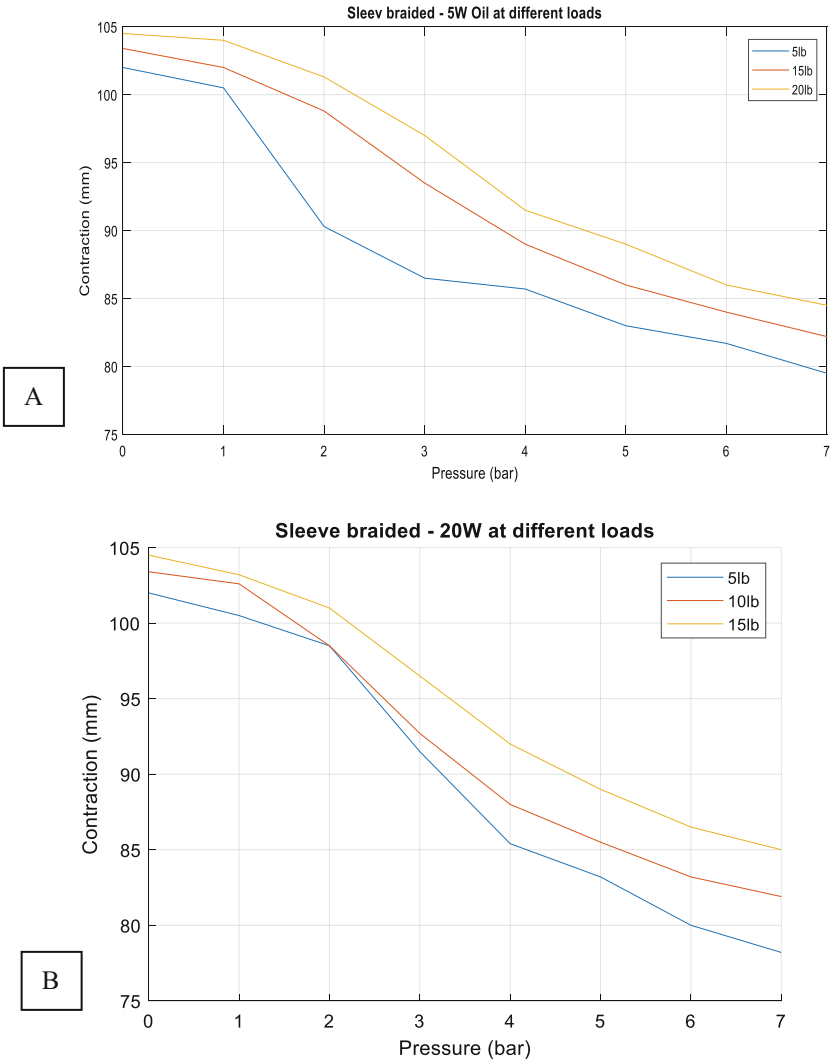


Fig. 13. Comparison of different fluids at varying loads.

The 5 W motor oil being the least viscous which is represented the graph in Fig. 13A has the best contractions as compared to the 20 W motor oil (Fig. 13B). The less viscous a fluid is, the better the flow rate is achieved through all joints and valve fixtures.

6 Design Concepts

The exoskeleton was designed with the user in mind. This was done by implementing many safety features which are incorporated into the electronic system. The exoskeleton should perform an automated lifting process using relevant sensors and control mechanisms. These features can be seen in the electronic and lifting design systems.

6.1 Electronic Design

The electronic design of the system makes use of several safety factors such as; a pressure cut-off switch which is built into the coding for the fluidic muscle and a rack and pinion set-up which works in conjunction with a potentiometer to determine the contraction of the muscle. Instead of using pressurized air as the driving fluid, an oil-based substance is used. This ensures the whole system is portable where a simple motor and linear actuator will act as the pump for the muscle itself as seen in Fig. 14. When a load is applied, a load cell is used to determine the weight being carried and engage an automated lifting system to a reference height that specified.

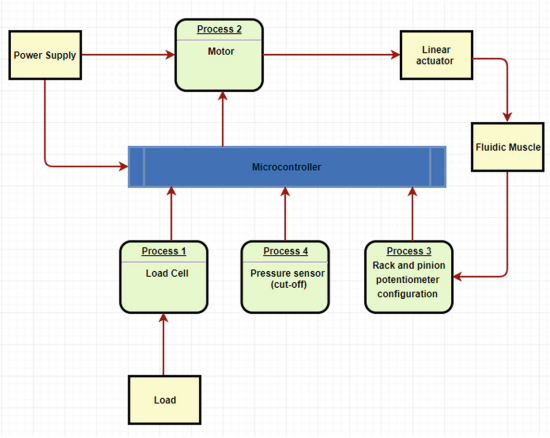


Fig. 14. Electronic design system.

6.2 Lifting Mechanism

The lifting mechanism comprises of pulleys, a bleeder valve screw, hydraulic accumulator, McKibben fluidic muscle and Bowden cables. Before an operation, a fluid (oil) is induced into the muscles. These muscles are open muscles where fluid travels through them and into the distribution plate and block. A bleeder valve screw is used to release all the air in the system. Another safety feature which is introduced into the lifting system is a hydraulic accumulator. This accumulator acts as a damper where it is used for any sudden shocks exerted onto the system (increased weight while carrying a load).

To effectively lift a load, a pulley system is used to transmit the force from the front towards the back. The pulleys are also used to acquire greater rotation of forearm by using a double pulley with different diameter sizes as seen in Fig. 15.

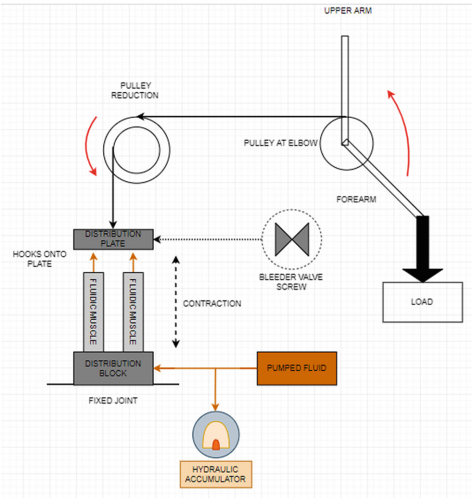


Fig. 15. Lifting mechanism.

7 Discussion

The research conducted on exoskeletons and artificial muscles provided information on how exoskeletons differ from other exoskeletons according to their specific application.

To design an exoskeleton arm suit that can be used to perform several different tasks, the study of the biology of the human arm is vital for flexible joint analysis and restricted motions due to those joints. By also considering the operation of the human muscles, it is evident how efficient the arm muscles are and that simulating these muscles can be advantageous.

The main purpose of this paper was to analyse the suitability of implementing a biologically inspired exoskeleton to enhance one's capability. This can be ascertained by considering the kinematic models of the desired exoskeleton design to simulate the motion of the human arm with imposed restrictions to further improve functionality whilst reducing the chance of injury. These models are simulated and analysed to understand the singularities that are currently present in the exoskeleton as seen in the workspace models.

Other support systems such as the electronic and biomechanical systems are required to assist the exoskeleton to perform operations that are portable and safe. Electronic cut-off operations are considered to assist with the safety of the user and the operating system. The biomechanical design allows for low space requirements of the system and also allows for portability.

Variations of fluid insertions were used to compare which fluid can provide the best contraction and was able to support large loads. From testing, it's apparent that 5 W motor oil displays the best contraction properties for fluidic muscles.

8 Conclusion

Although exoskeleton suits have been around for a while, these suits have not broken the bridge between biological and mechanical systems where both aspects are integrated to form an exoskeleton that supports an individual while using biomechanical devices such as Fluidic muscles that simulate actual muscles. These biologically inspired characteristics create the connection of man and machine forming this symbiotic relationship.

The other need for exoskeletons is for portable automated human enhancement systems which are currently under development, however, the cost factor involved with these systems are high. The problem with reducing the cost is selecting the actuation system that requires good strength with has lightweight compact bodies. By using Fluidic muscles, this is achieved. These muscles are easy to manufacture and have easy mounting configurations. The cost involved with exoskeletons is further reduced drastically by using oil as the driving fluid for all lifting applications instead of air operated systems.

References

1. Writer, S.: BusinessTech, 30 July 2019. https://businesstech.co.za/news/business/332169/south-african-unemployment-jumps-to-a-16-year-high-of-29/. Accessed 31 July 2019
2. Naidu, D.: Bio-mechatronic Implementation of a Portable Upper Limb Rehabilitative Exoskeleton. UKZN, Durban (2011)
3. Wang, B.: SuitX lowers cost of full body medical mobility exoskeleton to $40,000, 12 May 2018
4. Marinov, B.: Types of classifications of exoskeletons. Exoskeleton report (2015)
5. Rahman, M.H., Saad, M., Kenné, J.P., Archambault, P.S.: Modeling and control of a 7 DOF exoskeleton robot for arm. In: International Conference on Robotics and Biomimetics, Guilin, China (2009)
6. Drillis, R., Contini, R.: Body Segment Parameters. Office of Vocational Rehabilitation, New York (1966)
7. Almomani, A., Miqdadi, F., Hassanin, M., Samy, M., Awadallah, M.: The First Pneumatic Fluidic Muscles Based Exoskeleton Suit in the U.A.E. IEEE *(*2014)
8. Snazell, N.: Lumbar spine. Nicky Snazell, Stafford
9. Daerden, F., Lefeber, D.: Pneumatic artificial muscles: actuators for robotics and automation. Eur. J. Mech. Environ. Eng. **47**(1), 11–21 (2002)

The Applicability of Robotic Cars in the Military in Detecting Animate and Inanimate Obstacles in the Real-Time to Detect Terrorists and Explosives

Sara K. Al-Ruzaiqi(✉)

Computer Science Department, Higher College of Technology, Muscat, Oman
sara.alruzeiqi@gmail.com

Abstract. The most significant aspects of a robotic car are its ability to detect obstacles in real time and as well avoid such obstacles. This study, therefore, presents the design and implementation of a robotic car with real time obstacle detection and avoidance based on software, hardware and communication environments. The system's implementation has been founded on the android application, Arduino platform as well as the Bluetooth technology. Also presented in this study is the use of sensor programming in the design and application of a robotic car. It is through interactions with an android-based device that the robotic device has been successfully developed. On the other hand, the robot's brain is created from the Arduino Uno. There are numerous hardware components used to create the robot such as; the Bluetooth module, Buzzers, Ultrasonic sensor and the PIR sensor. Software components utilizing a mobile application are also constituents of the robot. The movement of the robotic car can be controlled by the user through selection of the desired direction or mode by mobile application. With the use of an intelligent device, the user can control the movements of the robotic car or on the other hand, switch to automatic mode where the car will drive on its own. Since the car has the real time ability to detect and avoid obstacles, it can flee from any obstacles on its path and as well detect live objects. The primary aim of this paper is to enlighten civilians and military on the substantial advantages of this technology. Its application on military grounds can play a significant role in detecting potential terrorist attacks through its live detectable sensors.

Keywords: Android-based devices · Robotic car based on Arduino · Integrated Development Environment · Obstacle detection · Obstacle avoidance

1 Introduction

Technological advancements over the last decade have resulted to application of sensors originally used in electronic devices in many more areas to improve a variety of life processes. Sensors are devices with the ability to convert a various energy forms into electrical energy. It is through sensors that the gap between various electronic devices and the environment is bridged. 'Environment' refers to any physical location such a

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 232–245, 2021.
https://doi.org/10.1007/978-3-030-55180-3_19

hospital, military grounds, airports, shopping malls and factories while on the other side; electronic devices could be tablets, robots, smartphones and smart clocks among others. In the industrial context, these devices are widely applied for purposes such as identification, controlling, imaging as well as security & protection. As of now, technological advancements have resulted to production of countless types of sensors such as heat sensors, pressure sensors, obstacle recognizers, and human detectors, among others. Whereas in the past sensors were predominantly used for lighting, their applications have evolved to make life easier. They are also remarkably attributed to incredibly fast developments in the electronics field. Consequently, this has opened a path for new inventions that have continuously made life easier. In the modern world, artificial intelligence algorithms are being applied to develop robot systems. Perception is the most crucial part of the robot, with a robot's design focused on its ability to perceive the environment. It is, for example, essential for a robot to be able to detect explosives or identify terrorists using sensors in a military area. It is crucial for a robot to have the ability to perceive variables, such as temperature change, in its immediate environment, interpret them and then act accordingly.

This article presents a remote and autonomously controlled robotic car and focuses on use of sensors to detect and avoid obstacles. The Bluetooth technology is used to create a connection between the robot and the android device. The Arduino Uno on the other hand processes incoming data, and based on the user's input value, a robotic action can be performed. An android application (mobile phone) is applicable in two main modes to control a robotic car. They are the automatic mode and the user control mode. Desired actions are selected from a menu with buttons displayed on the phone's screen. The buttons makes it possible to move the car forward and backward, turn it right or left, and switch between user control and automatic modes. In the automatic mode, the robotic car navigates itself without hitting any obstacles. It also has the potential to detect live beings and consequently give warnings once encountered. Temperature sensors enable the car to make live detections and signal the user with a red led light. This study embarks on a novel vehicle design empowered with real time obstacle detections and avoidance. No previous studies have been found to evidence past investigations on application of Arduino Uno and Android Platform to detect and avoid obstacles in real time.

Section 2, outlines literature from closely related studies while Sect. 3, lays down the working principles and system architecture of the robotic car. The specifications and materials used in the robotic car are explained in Sect. 4, and Sect. 5, entails its design and implementation. Section 6, is the conclusion to the article.

2 Related Works

This section is a comprehensive review of closely related past studies detailing on the working principles and methods therein that are relevant to this study. Families where all parents/guardians spent much time away were relieved when S. S. Pujari et al. [1] designed a Robot with the ability to remotely monitor children and consequently communicate through a camera. The main components of this robot were the Raspberry Pi 3, Wi-Fi, camera module and the Bluetooth technology. The robot's heart was defined as Raspberry Pi and was coded using Python language. In a study by M. R. Mishi et al. [2],

a robotic car jointly controlled by Arduino Uno and Raspberry Pi was designed. The car was traced using GPS, which was also applied in measuring the distance between the obstacle and the path. This model utilized cloud data without necessarily having to be online. This consequently derived control for the multi-motion system. In another study, D. Chakraborty et al. [3] applied sensors and Bluetooth technologies to design and develop a robotic car. In addition to establishing communication links between the robotic car and the smart device, the scientists also made it possible for living beings to be observed through the phone's camera. They also installed the ultrasonic ranging sensor that identified obstacles in the opposite direction and thus prevented occurrence of collisions. All images from the smart device's camera were recorded in a database and subsequently analyzed.

The military were supplied with a robot originally designed and developed in a study conducted by E. Amareswar et al. [4]. The robot was majorly used in detecting explosives with its installed metal detectors. It also gave the operators a real time view of the surrounding though the camera of the android device used. The main components of this robot were; an Android device, a microcontroller (Arduino Uno), metal detector, Bluetooth module, DC motors, wireless camera and motor driver.

Lastly, in a study conducted by Premkumar et al. [5], a robotic arm was designed and controlled using Raspberry Pi. This study primarily focused on adding features of the human arm to the robot's arm. Movements of the arm were facilitated by the Raspberry Pi code written in the Python language. It became possible for a user to move the arm in the desired direction using the android application whose code was written in Java. A Wi-Fi connection [5] was consequently developed as a means of communication between Raspberry Pi and the Android application. It is through this communication that the robotic arm was able to move right, left, up and down. This study has siphoned various methodologies and principles of operations from the above studies and combined it with own innovations to design and develop a remote and autonomously controlled robotic car. The car has the ability to detect and avoid obstacles in real time, its model is based on Arduino and Android application is used to command operations of the robotic car.

3 Rule-Based Preparation Work

As illustrated in Fig. 1, various sensors have been used to develop the structural system of the robotic car.

The robotic car in this study entails two major modes; the user control mode and the automatic mode. Bluetooth technology has been applied to establish communications between the robot and the android device.

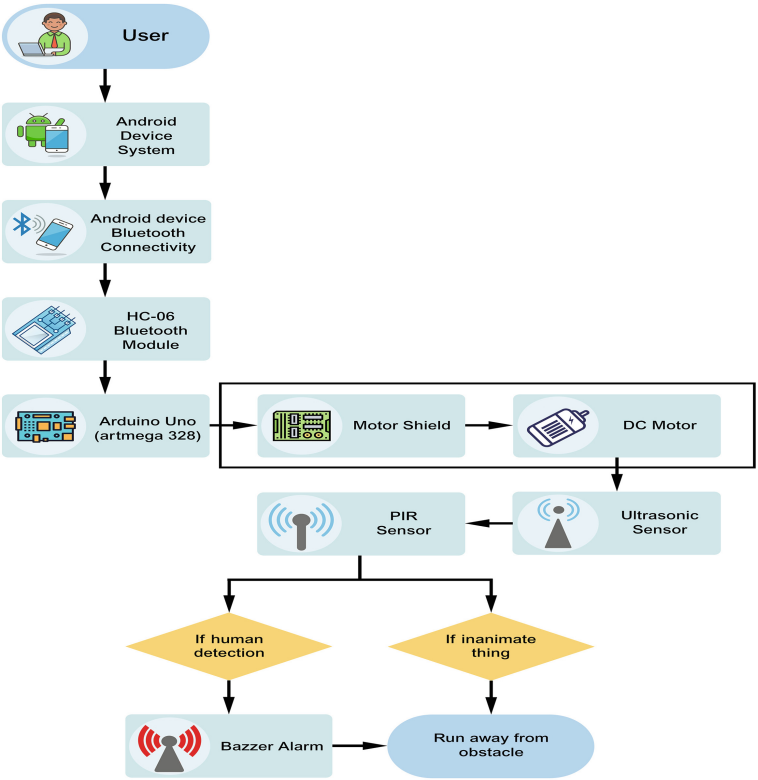


Fig. 1. An architecture module of robotic car.

4 Operational and Technical Requirements of the Robotic Car

4.1 Arduino Uno Board

The Arduino Uno, which has 14 pins and uses the ATmega328 microprocessor, forms the robot's brain. Shown in Fig. 2 is the Arduino UNO model, the most popular Arduino card.

This type of card is easily programmable with the use of Arduino libraries [6]. Its ease of programmability is its main advantage that makes it more preferable than other microprocessors. The Arduino Uno cannot be programmed in any environment other than the Integrated Development Environment (IDE) where Embedded C language is the programming language used. By utilizing signals from sensors, the Arduino Uno plays a vital role in designing and development of environment-sensitive robots and systems [7]. Consequently, the output of such robots and systems are specific to the immediate environment such as sound and light changes.

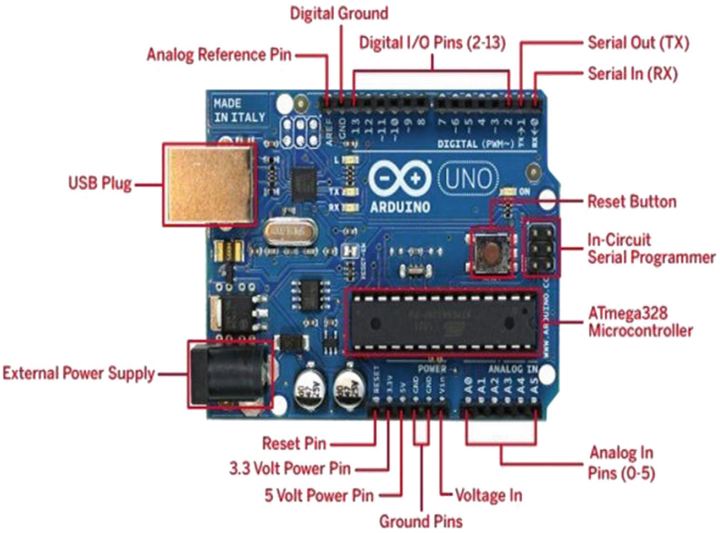


Fig. 2. Microcontroller board, Uno.

4.2 Bluetooth Transceiver Module Hc-06 for Arduino

Figure 3 is an illustration of the HC-06 Bluetooth Module responsible for providing communications between the devices at short distances of between 10–20 m. The module uses serial communication (USART) to facilitate communications with the Arduino [8]. It is not possible for the Bluetooth module to send connection requests to other modules as it can only respond to incoming requests. It comprises four pins; Tx, VCC, Rx, and GND.

Arduino supplies the GND and VCC to be used in the Bluetooth module [9]. For the module to detect commands emanating from the Arduino, the Tx pin from the Arduino must be plugged into the Bluetooth module's Rx part. It is only after this plug in has been done that the module can access messages from the Arduino. However, for the android device to be connected via the Bluetooth module, it is compulsory for a password to be set.

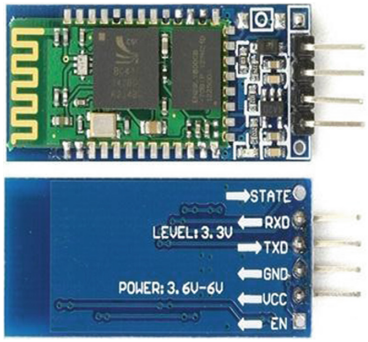


Fig. 3. HC-06 wireless module for Arduino.

4.3 Buzzer/Piezo Speaker

A buzzer enables production of various sound waves depending on the switched voltage. The buzzer is significantly lightweight, easy to produce, cheap and readily available in the market [10]. It is also largely used in a wide variety of appliances. The buzzer's main function is to warn users upon detection of a situation in a device/system. It therefore allows various inputs and emits warning sounds based on the inputs.

The buzzer's mode of operations begins with conversion of the DC voltage from the input port into oscillation signal, which is then amplified [10]. Amplification of a piezo-discrete high voltage results to mechanical expansion and contraction. The metal plate, in turn, bends in the opposite direction. Subsequently, constant twisting of the metal plate in the opposite direction [10], as shown in Fig. 4, causes the shrunk iceberg to release sound waves in the air.

Fig. 4. Passive buzzer module for Arduino.

4.4 Arduino Motor Shield

As illustrated in Fig. 5, Arduino Motor Shield is an L298-based motor driver. The card main functions are to provide control for the DC motor drive's speed and direction and as well measure its current.

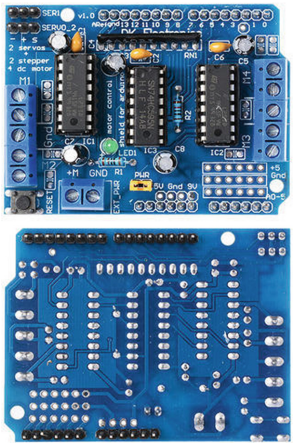


Fig. 5. Arduino motor driver shield.

4.5 DC Motor Speed Control

Conversion of direct current electrical energy into mechanical energy takes place in the Direct Current (DC) motor [11]. As an electric machine, the DC motor operates under the principle that "A current carrying conductor is exposed to humid when it enters a magnetic field." The applied force is calculated using the formula is $F = BIL$ [12].

A DC motor has six main constituents namely; brush, coils, direct current source, magnets, rotors and stator. Figure 6 is an illustration of the aforementioned parts. The purpose of the DC motor in this study is to turn the wheel. A DC motor generates mechanical force when direct current rotates the armature placed at the middle of a magnetic field generated by the coils.

Fig. 6. DC electric motor.

4.6 Arduino Motion Detector Using PIR Sensor

A Passive Infrared (PIR) Sensor is used to detect the temperatures of live beings. The term 'Passive Infrared' comes from its distinctive characteristic of not emitting any energy or heat. The sensor therefore, detects live beings by use of infrared rays. PIR sensors are characterized by low power consumption as well as low costs of operations. As illustrated in Fig. 7, there two pin slots in a PIR sensor, of which both are Infrared-sensitive [13]. It is with their heat spread that live beings such as human beings enters the sensor's field of view. Consequently, changes in the sensor's temperatures occur when the live being emits heat. This changes in turn leads to perception of the live being's movements.

Fig. 7. PIR motion sensor alarm.

4.7 Control HC-SR04 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor calculates the distance to an object by use of the Sound Navigation and Variable (Sonar) [14]. Since ultrasonic sensors emit ultrasonic sound waves, their application makes it possible to make distance measurements between the obstacle and the robotic car. The measurement accuracy level is higher in ranges of 1–400 cm between the obstacle and the robotic car [14]. The frequencies of ultrasonic sound waves ranges between 20 kHz and 500 kHz [14]. As illustrated in Fig. 8, the distance to an obstacle is determined by calculating the time taken by ultrasonic sound waves emitted by the ultrasonic sensor to hit the obstacle. Under optimum conditions, ultrasonic sensors can have a sensing range of up to 30 m [14]. An ultrasonic microphone and an ultrasonic speaker are the two transducers of ultrasonic sensors.

To calculate the distance between an obstacle and the sensor, an electronic circuit is engaged in determining the time spent to relay the sound wave from the ultrasonic loudspeaker until it hits the obstacle and is reflected back to the ultrasonic microphone [14]. The distance is then determined by dividing the recorded time with the speed of the sound wave. Figure 9 is a diagrammatic illustration of the process.

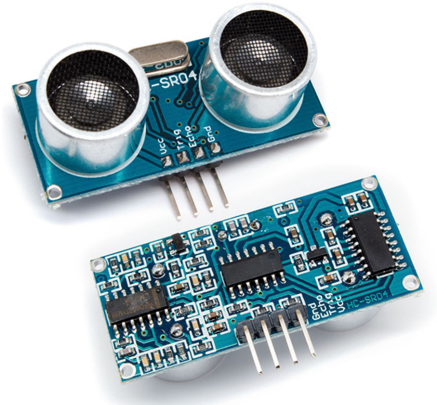


Fig. 8. Ultrasonic distance sensor.

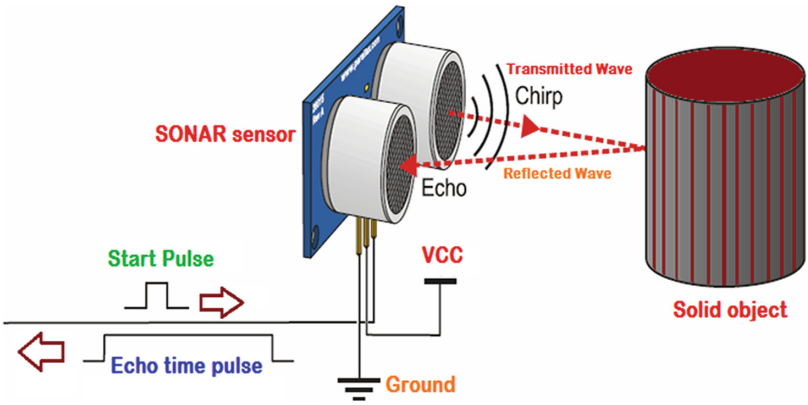


Fig. 9. Distance measurement using ultrasonic sensor

4.8 Arduino IDE

As a software development platform, the Arduino IDE facilitates usage of Arduino kits, writing of codes, compilation of the codes, and finally loading the derived and compiled codes into the Arduino Uno which is usually connected via the USB port of the user's computer. Arduino IDE uses C/C++ languages, which have two primary functions [10];

1. Setup – a function that commences working from the launch of the program
2. Loop – a function running in a loop for as long as the cards power is on and stops when the card power is off.

4.9 Create an Android Controlled Robot Using the Arduino Platform

Development of the android platform whose operating system is Linux-based is attributed to various developers such as the Open Handset Alliance and Google. The Linux-based

operating system, initially used for smartphones and tablets, has a distinctive open source code as well as low cost [15]. There are four main layers [15] in this type of android platform namely;

1. Linux Kernel.
2. Libraries and Android Runtime.
3. Application Framework.
4. Applications.

The first layer, Kernel, performs memory management, networking, and process management functions [15]. All libraries on the android OS are found in the second layer, Libraries & Runtime, and are usually written in C and C++ language. Java interface is used to invoke this layer, which also works with the Dalvik virtual machine. The second layer plays a vital role as a virtual translator for communications between the operating system and various applications [15]. The structure of an application for use by the android OS is determined by the third layer. Lastly, applications, the fourth layer, provide an interaction interface between users and java-written applications [15]. The connection between the robot and the application is founded on the Bluetooth technology.

5 Design and Implementation of the Robotic Car with Autonomous Obstacle Avoidance

As illustrated in Fig. 10 diagrammatic representation, the constituents of the robotic car are: Ultrasonic sensor HC-SR04, HC-06 Bluetooth module, Arduino Uno, Arduino motor shield, PIR sensor, 9 V battery, DC motor, and the buzzer. Figure 10 shows the Robotic car.

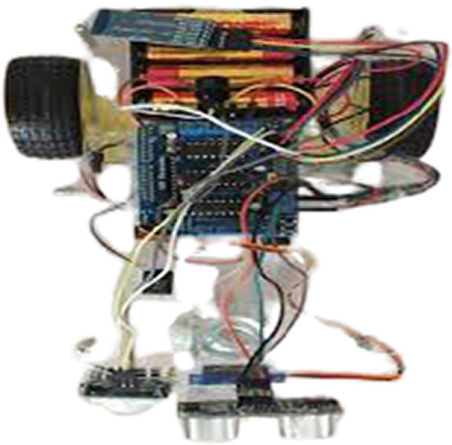


Fig. 10. Programming Arduino for obstacle avoiding robot.

Implementation commences with download of the Arduino Bluetooth Controller, which is available in the Google play store. After a successful download, the application is launched while making sure that the Bluetooth connection is open. The application is then connected to the HC-06 Bluetooth Module, with '1234' serving as the default password. Upon successful connection, values are assigned to the desired keys after which the robot sends input values.

The Bluetooth Module facilitates transfer of data to the Arduino Uno from the Android application. Incoming signals are controlled by the Arduino Uno, which also informs on the signals to be transmitted to the motor driver. Consequently, the inputs entered determine the order of movements for the robotic car.

An intelligent device, basically a smartphone, facilitates user control for the basic robot's movements, that is, forward and backward, right and left turns, and to stop motion. The intelligent device also allows switching into auto mode where the robotic car can drive itself. With the use of sensors, the robotic car is able to detect obstacles in the real time and as well determine whether they are live beings or not. When an obstacle is detected, the red led lights turn on, an alarm rings from the buzzer, and the shortest avoidance distance is calculated and the car proceeds accordingly. This is the case for both animate and inanimate obstacles. When the robot drives to a cliff, an abyss is perceived and the car stops.

5.1 GUI of the Automatic Control Mode

To use either mode, it all commences with download of the Arduino Bluetooth Controller, which is available in the Google play store. After a successful download, the application is launched while making sure that the Bluetooth connection is open. The application is then connected to the HC-06 Bluetooth Module, with '1234' serving as the default password.

Fig. 11. Graphical user interface.

On the Android application, the user can change buttons as illustrated in Fig. 11. Desired values can be assigned to various buttons when the symbol in the upper right corner is clicked as shown in the Fig. 11 The program immediately starts running when the 'start' button is clicked, subsequently enabling the user to select a desired action via the 'select' button.

Figure 12 is an illustration of assigned values and options in this study. There are five primary options used to control the robotic car's basic movement. The robot controls itself when the user selects the last option by pressing the 'automatic mode' button. While in this mode, the robotic car gives a warning signal when it detects living beings, and as well evades obstacles encountered in its way.

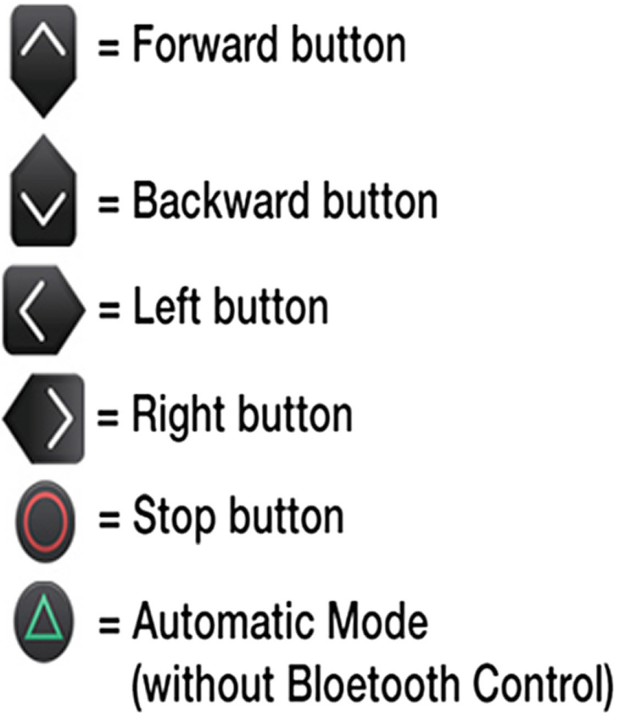


Fig. 12. GUI of the automatic control mode

In the manual mode, an android application enables the user to control the robot's basic movements. Through the Bluetooth technology, inputs entered by the user are forwarded to the robot. When inputs from the android device are received by the robot, they are processed in the Arduino Uno, the processor, and subsequent movement commands are relayed based on the input order. The main movement commands are forward and backward, right and left, as well as the stop function.

On the other hand, the robotic car can be switched to automatic mode by pressing the 'Automatic mode' key displayed on the android device. While in this mode, the robot controls its own basic movements. The ultrasonic sound enables the robot to detect

any obstacles that are within the range of 25 cm. The robot immediately stops upon detecting obstacle and reverses for 2 cm. The robot's system then activates the PIR sensor to determine whether the detected obstacle is live or not.

The PIR sensor distinguishes live obstacles by computing heat radiated from the obstacles. If the object moves and/or emits heat, the robot perceives it as animate and gives a warning in form of a red led light and an audible alarm sound from the buzzer. The robot then evades the live obstacle and continues on its way. On the other hand, when an obstacle is detected and perceived as an inanimate object by the robotic car, the furthest distance of evasion from the obstacle is calculated and the robot embarks on it towards the determined direction. This process is continuous until the user decides to stop the robot and exit the application, or until the robot's power supply is drained. In this mode, it is also possible for the buttons in the application interface to be configured based on user preference. The user can as well assign desired values to the buttons.

In this study, path tracking is formulated either by the robotic car itself while in the automatic mode or by the user while in the manual mode. When option 1 is selected (manual mode), the car's movements depend on the user's commands and enact real time obstacle avoidance. When in option 2 (Automatic mode), the robotic car employs the autonomous mode to detect and avoid obstacles in real time when it is on movement.

6 Conclusion

Whereas numerous studies have been conducted focusing on designing and developing of robotic cars based on Android platform, Arduino Uno, and Raspberry Pi technologies, no studies have been conducted to investigate remote and autonomous controlled robotic car based on the Arduino Uno and the Android platform. This study purposes to outline the applicability of robotic cars founded on Android applications in detecting animate and inanimate obstacles in the real time, and thus potential use in military grounds to detect terrorists and explosives. The basic movements of the robot are controlled through inputs from an android device application. The HC-06 Bluetooth module facilitates communications between the application and the robot. Crashing of the robot is prevented through the ultrasonic sensor HC-SR04, which enables real time detection and escape from obstacles and cliffs.

The robot is also able to distinguish between live and non-live obstacles by use of the PIR sensor. When a live being is detected, an audible alarm triggered by the buzzer rings. The novelty of this study is incorporation of the robot's ability to detect and avoid obstacles, both animate and inanimate, in the real time. As such, when such a robot is commissioned, the effectiveness of operations can be highly improved by its remote, wireless and autonomous control. More advanced and sophisticated materials can be used to improve this study. One viable recommendation is the substitution of the Bluetooth module with the Wi-Fi module that is not easily broken and has a larger distance range of connection.

References

1. Pujari, S.S., Patil, M.S., Ingleshwar, S.S.: Remotely controlled autonomous robot using Android application. In: 2017 IEEE International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud) (I-SMAC) (2017)
2. Mishi, M.R., Bibi, R., Ahsan, T.: Multiple motion control systems of the robotic car based on IoT to produce cloud service. In: 2017 IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE) (2017)
3. Chakraborty, D., Sharma, K., Roy, R.K., Singh, H., Bezboruah, T.: Android application based monitoring and controlling of movement of a remotely controlled robotic car mounted with various sensors via Bluetooth. In: 2016 IEEE International Conference on Advances in Electrical, Electronic and Systems Engineering (2016)
4. Amareswar, E., Goud, G.S.S.K., Maheshwari, K.R., Akhil, E., Aashraya, S., Naveen, T.: Multipurpose military service robot. In: 2017 IEEE International Conference of Electronics, Communication, and Aerospace Technology (ICECA) (2017)
5. Premkumar, K., Gerard Joe Nigel, K.: Smart phone-based robotic arm control using Raspberry Pi, Android, and Wi-Fi. In: 2015 IEEE International Conference on Innovations in Information, Embedded, and Communication Systems (ICIIECS) (2015). https://doi.org/10.1109/iciiecs.2015.7192973
6. Varshney, S., Gaur, B., Farooq, O., Khan, Y.U.: Brain machine interface for wrist movement using robotic arm. In: IEEE 16th International Conference on Advanced Communication Technology (2014). https://doi.org/10.1109/icact.2014.6779014
7. Cameron, N.: Arduino Applied: Comprehensive Projects for Everyday Electronics. Apress Publishing, New York (2018)
8. Dey, N., Mukherjee, A.: Embedded Systems and Robotics with Open Source Tools. CRC Press, Boca Raton (2018)
9. Internet: Geleceği yazanlar, Bluetooth ile İletişim. https://gelecegiyazanlar.turkcell.com.tr/konu/arduino/egitim/arduino-201/bluetooth-ile-iletisim. Accessed 07 Nov 2019
10. Karrupusamy, K., Chen, J., Shi, Y.: Sustainable Communication Networks and Application: ICSCN 2019. Springer, New York (2019)
11. Verma, S.: Android app controlled Bluetooth robot. Int. J. Comput. Appl. **152**(9), 35–40 (2014)
12. Internet: Robotpark, DC Motor Nedir?, 21 June 2018 http://www.robotpark.com.tr/Dc-Motor-Nedir
13. Internet: Güvenlik Online, PIR Sensör, https://www.guvenlikonline.com/makale/219/pir-sensor.html. Accessed 21 Oct 2019
14. Hadžikadić, M., Avdaković, S.: Advanced Technologies, Systems, and Applications II: Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT). Springer, New York (2018)
15. Barry, P., Crowley, P.: Modern Embedded Computing: Designing Connected, Pervasive, Media-Rich Systems. Elsevier, London (2012)

Synthesis of Control System for Quad-Rotor Helicopter by the Network Operator Method

Askhat Diveev[1,2(✉)], Oubai Hussein[2], Elizaveta Shmalko[1], and Elena Sofronova[1]

[1] Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Vavilova str., 44, Moscow 119333, Russia
aidiveev@mail.ru
[2] RUDN University, Miklukho-Maklaya str., 6, Moscow 117198, Russia

Abstract. The control synthesis problem for a complex object of large dimension is considered. In the task, it is necessary to stabilize the object relative to the state space point. To automate the solution of synthesis problem, it is proposed to use the network operator method. Description of the network operator method is presented briefly. As an example, the synthesis of a quad-rotor helicopter stabilization system is considered. At the synthesis a decomposing technology of the mathematical model of control object is used to reduce the dimension of the system. Firstly, the synthesis of control system for angular movement is made. Then, a control system for spatial stabilization is synthesized. As a result, mathematical expressions for describing of the quad-rotor helicopter control system are found by the network operator method.

Keywords: Synthesis of control · Network operator method · Quad-rotor helicopter

Introduction

A control synthesis problem belongs to a class of problems for which an effective computational method has not yet been created because the solution of the problem is a mathematical expression of a multidimensional control function with a states multidimensional vector as arguments. Whenever a control synthesis problem is solved, a specific model of the control object is considered and based on the analysis of this model, a method for solving of the synthesis problem is proposed.

Attempts to analytically solve the synthesis problem are obviously possible only for very simple mathematical models of control objects, and in general case are doomed to failure. The same problems arise when solving any differential or algebraic equation. Analytical solutions can only be found for simple equations. For complex equations, numerical methods are used. But for these problems at least, there are sufficiently effective numerical methods. These numerical

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 246–263, 2021.
https://doi.org/10.1007/978-3-030-55180-3_20

methods can be used for these problems described in a general form. A feature of the synthesis problem consists in the fact that there isn't a numerical method for its solution in general form.

Since a solution of the control synthesis problem is a mathematical expression of the control function, a set of possible solutions belongs to the space of formulas, which does not have a numerical measure between any two possible elements. In regression tasks the desired mathematical expression is often written with accuracy up to the parameters values. Note that an artificial neural network can be described also by a mathematical expressions with accuracy to parameters values. The problem of control synthesis can be interpreted as changing the control object mathematical model in order to give the system of differential equations new qualitative properties, for example, property of stability relative to some point in the state space. Obviously, to synthesize a control system that provides stable property for a control object, Lyapunov's method can be used. But this approach requires constructing Lyapunov function [1,2]. But constructing Lyapunov function for any nonlinear system of differential equations is a very difficult task. Here we have to construct Lyapunov function in general form for any mathematical model of control object.

In the middle of the 20th century, a fairly general method of analytical design of optimal controllers (ADOC) is appeared. The method allowed the calculation of a multivariate linear feedback regulator. Basically the method was designed for linear systems with a quadratic quality criterion [3]. Further development of the ADOC method on its application to nonlinear control systems in the end of the last century led to the creation of the method of analytical design of aggregate controllers (ADAC) [4]. This method is difficult for automated computing. To implement the method, one must analytically resolve equations for aggregated variables and find a control function, at this the number of equations can be bigger than the dimension of the control vector. Both methods are not universal, as far as analytical solutions are obtained depending on the specific object model and terminal manifolds.

In this work for the control system synthesis problem the network operator method is used [5]. It belongs to the class of symbolic regression methods. They were developed for the task of automatic programs writing. Symbolic regression techniques are looking for a possible solution in the form of code by a genetic algorithm. There are now more than ten methods of symbolic regression. They all differ in the form of coding. To apply the genetic algorithm depending on the type of code, special operations of the genetic algorithm crossover and mutation have been developed. Some methods of symbolic regression use the principle of small variations of a basic solution. This principle is a universal approach to create search algorithms for optimal solution on non-numerical space [6]. Any method of symbolic regression can be used for the control system synthesis. Application of these methods allows to automate the process of synthesis and to obtain mathematical expressions for control system [7].

In this work we apply the network operator method for control system synthesis of quad-rotor helicopter.

1 Problem Statement of Optimal Control Synthesis

Let us consider formal statement of the optimal control synthesis problem. A mathematical model of the control object in the form of the system of ordinary differential equations is given

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{1}$$

where $\mathbf{x}$ is a vector of a states' space, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} = [x_1 \ \dots \ x_n]^T$, $\mathbf{u}$ is a vector of control, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$, U is a compact set, $m \leq n$.

A domain of initial conditions is given

$$\mathrm{X}_0 \subseteq \mathbb{R}^n. \tag{2}$$

The terminal conditions are set

$$\mathbf{x}^f \in \mathbb{R}^n. \tag{3}$$

The quality criterion in the form of integral functional is given

$$J = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \to \min, \tag{4}$$

where t_f is an end time of the control process. This time is not set, but is limited and is defined by a moment when terminal conditions are achieved

$$t_f = \begin{cases} t, \text{ if } t < t^+ \text{ and } \left\|\mathbf{x}(t) - \mathbf{x}^f\right\| \leq \varepsilon_1 \\ t^+ - \text{ otherwise} \end{cases}, \tag{5}$$

t^+ and ε_1 are set positive values.

It is necessary to find a control function in the form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}) \tag{6}$$

where

$$\mathbf{h}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m, \tag{7}$$

$$\mathbf{h}(\mathbf{x}) \in \mathrm{U}, \ \forall \mathbf{x} \in \mathbb{R}^n, \tag{8}$$

if this function $\mathbf{h}(\mathbf{x})$ to insert to mathematical model of control object (1), then the system of differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x})) \tag{9}$$

will have for any initial conditions from the domain (2), $\mathbf{x}(0) \in \mathrm{X}_0$ such partial solution $\mathbf{x}(t, \mathbf{x}(0))$ that will hit to the terminal condition (3) for limited time t^+ with optimal value of the quality criterion (4)

$$J = \int_0^{t_f} f_0(\mathbf{x}(t, \mathbf{x}(0)), \mathbf{h}(\mathbf{x}(t, \mathbf{x}(0)))) dt. \tag{10}$$

This means in ideal case, that the value of the criterion coincides with the value of the same criterion for the optimal control problem, when in the problem (1), (3)–(5) for one initial condition $\mathbf{x}(0) \in \mathrm{X}_0$ a solution is looked for in the form of a function of time

$$\mathbf{u} = \mathbf{v}(t). \tag{11}$$

According to the problem statement in the result of solving the optimal control synthesis problem the terminal point (3) has to become a stable equilibrium point in the state space at least for the domain of initial conditions (2).

As the model (1) is stationary and does not depend on time, it is more convenient to find a solution to the synthesis problem (1)–(5) in a form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^* - \mathbf{x}), \tag{12}$$

where $\mathbf{x}^*$ is some goal point in the state space.

Then after solving the synthesis problem the point $\mathbf{x}^*$ can always be exchange onto the terminal point $\mathbf{x}^f$ (3).

To solve the synthesis problem the network operator method is applied.

2 The Network Operator Method

The network operator method codes a mathematical expression as a composition of elementary functions in the form of an oriented graph.

The method uses following basic sets.

The first one is the set of arguments of the mathematical expression

$$\mathrm{F}_0 = (f_{0,1} = \Delta x_1, \ldots, f_{0,n} = \Delta x_n, f_{0,n+1} = q_1, \ldots, f_{0,n+p} = q_p), \tag{13}$$

where

$$\Delta x_i = x_i^* - x_i, \ i = 1, \ldots, n, \tag{14}$$

x_i^* is a component of the goal point $\mathbf{x}^* = [x_1^* \ldots x_n^*]^T$, $i = 1, \ldots, n$, q_i is a component of the parameter vector $\mathbf{q} = [q_1 \ldots q_p]^T$, $i = 1, \ldots, p$.

Components of the parameter vector has been included as arguments of the mathematical expression to expand the functions space. The optimal value of the parameter vector is found together with the structure of the mathematical expression.

The second one is the set of functions with one argument

$$\mathrm{F}_1 = (f_{1,1}(z) = z, f_{1,2}(z), \ldots, f_{1,W}(z)), \tag{15}$$

where $f_{1,1}(z)$ is an identity function.

The third one is the set of functions with two arguments

$$\mathrm{F}_2 = (f_{2,1}(z_1, z_2), \ldots, f_{2,V}(z_1, z_2)). \tag{16}$$

Functions with two arguments have to be commutative

$$f_{2,i}(z_1, z_2) = f_{2,i}(z_2, z_1), \ i = 1, \ldots, V, \tag{17}$$

and associative

$$f_{2,i}(z_1, f_{2,i}(z_2, z_3)) = f_{2,i}(f_{2,i}(z_1, z_2), z_3),\ i := 1, \ldots, V, \tag{18}$$

and to have unit element

$$f_{2,i}(z, e_i) = f_{2,i}(e_i, z) = z, \tag{19}$$

where e_i is a unit element of a function $f_{2,i}(z_1, z_2)$, $i = 1, \ldots, V$.

When writing a function with two arguments and with unit element as one of arguments, the unit element is not written. Therefore the function with two arguments is written like a function with one argument

$$f_{2,i}(e_i, z) = f_{2,i}(z, e_i) = f_{2,i}(z),\ i \in \{1, \ldots, V\}. \tag{20}$$

Associative property of functions with two arguments allows to use them for calculations with any number of arguments

$$f_{2,i}(z_1, \ldots, z_k) = f_{2,i}(z_1, f_{2,i}(z_2, f_{2,i}(z_3 \ldots, z_k) \ldots)). \tag{21}$$

To code the mathematical expression in the form of a network operator it is necessary to write down this mathematical expression in the form of composition of elements from the sets (13)–(16).

To construct the network operator graph the following rules are applied: arguments of the mathematical expression are associated with source-nodes of the graph, functions with one argument are associated with arcs of the graph, and functions with two arguments are associated with nodes of the graph.

In a computer memory the network operator graph is presented as an integer upper triangular matrix. Diagonal element of the matrix can be zero or the number of the function with two arguments. Non diagonal element is the number of function with one argument. The network operator matrix is constructed from an adjacency matrix of oriented graph. When searching for the mathematical expression in the form of the network operator a decoding operation from the integer network operator matrix to mathematical expression should be used.

Proposition 1. *If integer upper triangular $L \times L$ matrix $\mathbf{\Psi} = [\psi_{i,j}]$, $i, j = 1, \ldots, L$ has some zeros on the main diagonal in first lines and the function numbers with two arguments on the main diagonal in other lines and non-diagonal elements are equal zero or the function number with one argument and there is at least one nonzero non-diagonal element in each line and column of the matrix, then this matrix is a code of a mathematical expression.*

Proof. For decoding the network operator matrix next formulas are used

$$z_j^{(i)} \leftarrow \begin{cases} f_{2,\psi_{j,j}}(z_j^{(i-1)}, f_{1,\psi_{i,j}}(z_i^{(i-1)})), \text{ if } \psi_{i,j} \neq 0 \\ z_j^{(i-1)} - \text{otherwise} \end{cases}, \tag{22}$$

where $i = 1, \ldots, L-1$, $j = i+1, \ldots, L$,

$$z_j^{(0)} = \begin{cases} f_{0,k} \in \mathrm{F}_0, \text{ if } \psi_{j,j} = 0 \\ e_{\psi_{j,j}} - \text{ otherwise} \end{cases} \tag{23}$$

Calculations of (23) lead to obtain a vector $\mathbf{z}^{(0)} = [z_1^{(0)} \ldots z_L^{(0)}]^T$, where every component is equal to an element of the set F_0 or a unit element of the function with two arguments z. According to (22) a nonzero element $\psi_{i,j} \neq 0$ of the matrix $\mathbf{\Psi}$ changes value of the component $z_j^{(i)}$ of the function with two arguments. Decoding an integer matrix $\mathbf{\Psi}$ using (22) and (23) results in a vector whose components are equal to either the mathematical expression or arguments of the mathematical expression. There are no other options. □

To find the optimal mathematical expression in the form of the network operator a variation genetic algorithm is used. According to the algorithm, one basic possible solution is codded in the form of integer network operator matrix. Other possible solutions are codded by the sets of variation vectors. The variation vector includes four components and describes one small variation of the network operator matrix, for example it changes one element of the network operator matrix. All genetic operations are performed on the sets of variation vectors. Thus, a variation genetic algorithm searches for solutions in the neighborhood of the given basic solution. After several generations, the epoch is changed and the basic solution is replaced by the best current possible solution. More details on the network operator method can be found in [5,8].

3 The Synthesis of Control System for Quad-Rotor Helicopter

Let us show how the synthesis problem is solved by the network operator method for multidimensional control object. The system of 12 differential equations is considered

$$\begin{aligned} \dot{x}_1 &= x_4 + (x_5 \sin(x_1) + x_6 \cos(x_1)) \sin(x_2)/cos(x_2), \\ \dot{x}_2 &= (x_5 \sin(x_1) + x_6 \cos(x_1))/\cos(x_2), \\ \dot{x}_3 &= x_5 \cos(x_1) + x_6 \sin(x_1), \\ \dot{x}_4 &= x_5 x_6 (I_2 - I_3)/I_1 + u_1/I_1, \\ \dot{x}_5 &= x_4 x_6 (I_3 - I_1)/I_2 + u_2/I_2, \\ \dot{x}_6 &= x_4 x_5 (I_1 - I_2)/I_3 + u_3/I_3, \end{aligned} \tag{24}$$

$$\begin{aligned} \dot{x}_7 &= x_{10}, \\ \dot{x}_8 &= x_{11}, \\ \dot{x}_9 &= x_{12}, \\ \dot{x}_{10} &= u_4 \sin(x_3) \cos(x_2) \cos(x_1) + \sin(x_1) \sin(x_2), \\ \dot{x}_{11} &= u_4 \cos(x_3) \cos(x_1) \cos(x_2) - g, \\ \dot{x}_{12} &= u_4 \cos(x_2) \sin(x_1) - \cos(x_1) \sin(x_2) \sin(x_3), \end{aligned} \tag{25}$$

where Eqs. (24) describe an angular movement, and Eqs. (25) describe a spatial movement, x_1, x_3 are rotation angles about horizontal axes, x_2 is a rotation angle about vertical axis, x_4 and x_6 are angular speeds of rotation about horizontal axes, x_5 is an angular speed of rotation about vertical axis, x_7, x_9 are horizontal axes, x_8 is a vertical axis, x_{10} is a speed along axis x_7, x_{11} is a speed along axis x_8, x_{12} is a speed along axis x_9, u_i is a moment around axis x_i, $i = 1, 2, 3$, u_4 is a total lift of all four screw, g is the acceleration of gravity, I_i is a moment of inertia around axis x_i, $i = 1, 2, 3$.

The quad-rotor helicopter coordinate system is shown in Fig. 1. The variables are simply numbered, without reference to their physical meaning, since the computational approach that is used to solve the synthesis problem is automatic and the computer does not care what these or other variables physically mean.

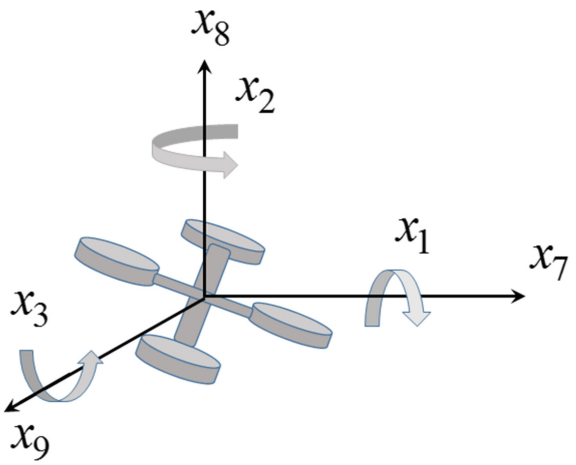


Fig. 1. The coordinate system of quad-rotor helicopter

To solve the synthesis problem and to achieve the object's stability in the twelve-measured state space, two problems are solved consequently, the synthesis of an angular and a spatial stabilization systems.

For the first problem the system (24) is used. It is necessary to find the optimal control in the following form

$$u_i = h_i(x_i^* - x_i), \ i = 1, \ldots, 6, \tag{26}$$

where x_i^* is a given coordinate i of a point in the six-measured space $\{x_1, \ldots, x_6\}$, $i = 1, \ldots, 6$.

In the problem of angular stabilization the following initial conditions are used

$$\begin{aligned}\mathrm{X}_0 = \{&\mathbf{x}^{0,1} = [-0.2\ -0.2\ -0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,2} = [-0.2\ -0.2\ 0\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,3} = [-0.2\ -0.2\ 0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,4} = [-0.2\ 0\ -0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,5} = [-0.2\ 0\ 0\ 0\ 0\ 0]^T, \mathbf{x}^{0,6} = [-0.2\ 0\ 0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,7} = [-0.2\ 0.2\ -0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,8} = [-0.2\ 0.2\ 0\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,9} = [-0.2\ 0.2\ 0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,10} = [0\ -0.2\ -0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,11} = [0\ -0.2\ 0\ 0\ 0\ 0]^T, \mathbf{x}^{0,12} = [0\ -0.2\ 0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,13} = [0\ 0\ -0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,14} = [0\ 0\ 0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,15} = [0\ 0.2\ -0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,16} = [0\ 0.2\ 0\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,17} = [0\ 0.2\ 0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,18} = [0.2\ -0.2\ -0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,19} = [0.2\ -0.2\ 0\ 0\ 0\ 0]^T, \mathbf{x}^{0,20} = [0.2\ -0.2\ 0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,21} = [0.2\ 0\ -0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,22} = [0.2\ 0\ 0\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,23} = [0.2\ 0\ 0.2\ 0\ 0\ 0]^T, \mathbf{x}^{0,24} = [0.2\ 0.2\ -0.2\ 0\ 0\ 0]^T,\\ &\mathbf{x}^{0,25} = [0.2\ 0.2\ 0\ 0\ 0\ 0]^T, \mathbf{x}^{0,26} = [0.2\ 0.2\ 0.2\ 0\ 0\ 0]^T, \}\end{aligned}$$

The terminal conditions was

$$\mathbf{x}^* = [0\ 0\ 0\ 0\ 0\ 0]^T.$$

Restriction on the control were

$$-2 = u_i^- \le u_i \le u_i^+ = 2,\ i = 1, 2, 3.$$

The quality criterion was

$$J = \sum_{i=1}^{26} \left(t_{f,i} + a_1 \sqrt{\sum_{j=1}^{6} (x_j^* - x_j(t_{f,i}, \mathbf{x}^{0,i}))^2} \right), \tag{27}$$

where a_1 is a weight coefficient, $a_1 = 1$, $t_{f,i}$ is a terminal time for solution with initial condition $\mathbf{x}^{0,i}$, a terminal time is defined by Eq. (5) with $t^+ = 1.5$, $\varepsilon_1 = 0.01$, $t^+ = \mathbf{x}(t, \mathbf{x}^{0,i})$ is a partial solution of the system (24) with control (26) and initial conditions $\mathbf{x}^{0,i}$, $i \in \{1, \ldots, 26\}$.

When searching for the optimal solution, parameters of the model were $I_1 = 1.5$, $I_2 = 1$, $I_3 = 1.5$, $g = 9.8067$. The network operator method had the following parameters: a dimension of the network operator matrix 32×32, a number of function with one argument $W = 20$, a number of function with two arguments $V = 2$, a number of possible solution in initial set $H = 1024$, a number of generation $P = 128$, a number of crossovers in one generation $R = 128$, a number of generation between change of the basic solution $E = 32$, a number of parameters $p = 6$. The basic solution was

$$u_i(0) = \sum_{i=1}^{6} q_i (x_i^* - x_i),\ i = 1, 2, 3. \tag{28}$$

In the result of solving the synthesis problem by the network operator method, the following solution was obtained

$$u_i = \begin{cases} u_i^-, \text{ if } \tilde{u}_i < u_i^- \\ u_i^+, \text{ if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i - \text{ otherwise} \end{cases}, \ i = 1, 2, 3, \tag{29}$$

Mathematical expressions for control function were the following:

$$\begin{aligned} \tilde{u}_1 = & \left(q_4(x_4^* - x_4) + q_1(x_1^* - x_1) + (x_4^* - x_4)^3 + \sqrt[3]{q_1(x_1^* - x_1)}\right)^{-1} \\ & + \left(q_4(x_4^* - x_4) + q_1(x_1^* - x_1) + (x_4^* - x_4)^3 + \sqrt[3]{q_1(x_1^* - x_1)}\right)^{1/3} \\ & + \operatorname{sgn}(x_6^* - x_6) \log(|q_6(x_6^* - x_6)| + 1) + q_2(x_2^* - x_2) \\ & + \operatorname{sgn}(x_4^* - x_4)\sqrt{|q_4(x_4^* - x_4)|} + q_1(x_1^* - x_1) + (q_4(x_4^* - x_4))^3, \end{aligned} \tag{30}$$

$$\begin{aligned} \tilde{u}_2 = & \operatorname{sgn}(\operatorname{sgn}(A + q_3(x_3^* - x_3) + x_3^* - x_3) \\ & \times \exp(|A_1 + q_3(x_3^* - x_3)x_3^* - x_3|) - 1) \\ & \times (|\operatorname{sgn}(A_1 + q_3(x_3^* - x_3) + x_3^* - x_3) \\ & \times \exp(|A_1 + q_3(x_3^* - x_3)x_3^* - x_3|) - 1)|)^{1/2} \end{aligned} \tag{31}$$

where $q_1 = 12.224$, $q_2 = 14.197$, $q_3 = 13.611$, $q_4 = 4.361$, $q_5 = 9.989$, $q_6 = 4.144$,

$$A_1 = \operatorname{sgn}(x_5^* - x_5) \log(|q_5(x_5^* - x_5)| + 1).$$

$$\begin{aligned} \tilde{u}_3 = & \tanh(0.5B_1) - \operatorname{sgn}(x_5^* - x_5) \log(|q_5(x_5^* - x_5)| + 1) \\ & - q_3(x_3^* - x_3) - x_3^* + x_3 + \sqrt[3]{B_1} + q_6(x_6^* - x_6) + q_2(x_2^* - x_2), \end{aligned} \tag{32}$$

where

$$\begin{aligned} B_1 = & (\operatorname{sgn}(x_5^* - x_5) \log(|q_5(x_5^* - x_5)| + 1) + q_3(x_3^* - x_3))^3 \\ & + q_6(x_6^* - x_6) + q_2(x_2^* - x_2), \end{aligned}$$

$$\tanh(\alpha) = \frac{1 - \exp(-2\alpha)}{1 + \exp(-2\alpha)}.$$

Figure 2 and 3 show the trajectories of helicopter movement on the vertical plane from eight initial conditions.

At the searching process for the optimal solution using the network operator method, the calculation of the functional and simulations of the model was 1 035 720 times. Calculations are performed on the computer with processor Core i7, 2.8 GHz. The calculation time was about 40 min.

Trajectories of helicopter movement show that the received synthesized control system stabilizes the object relative the terminal point. The quad-rotor helicopter moves from any initial conditions to the given terminal condition with the optimal value of the criterion (27).

In the Fig. 4, 5 and 6 the plots of control are presented for initial condition

$$\mathbf{x}^{0.1} = [-0.2 \ -0.2 \ -0.2 \ 0 \ 0 \ 0]^T.$$

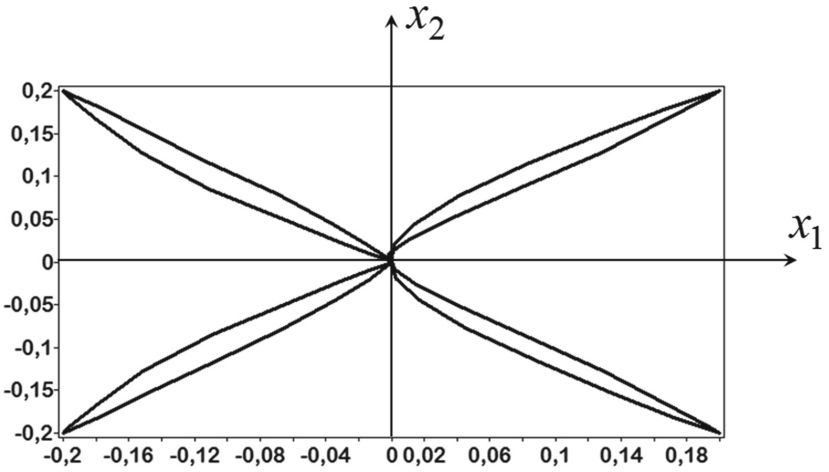


Fig. 2. Trajectories on a plane $\{x_1, x_2\}$ from eight initial conditions

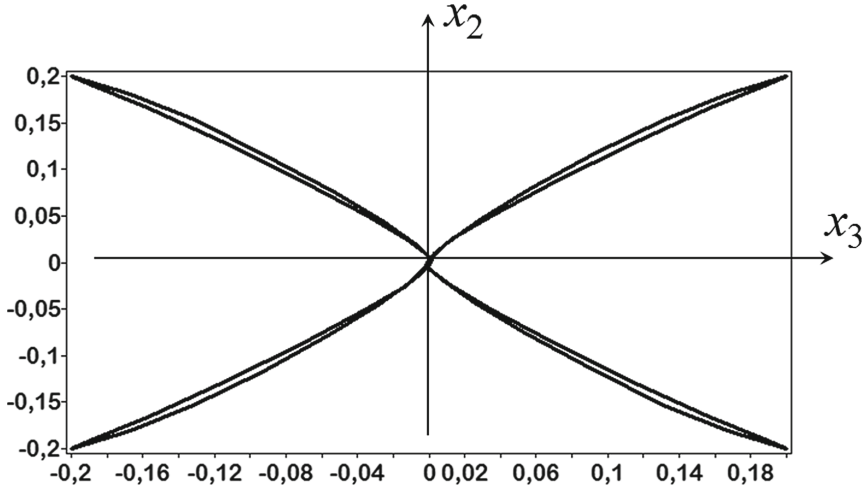


Fig. 3. Trajectories on a plane $\{x_3, x_2\}$ from eight initial conditions

Plots show that control during stabilization process does not violate restrictions.

On the second stage the full dynamic model of quad-rotor helicopter is considered. The angular model is replaced by the model with the stabilization system (30)–(32).

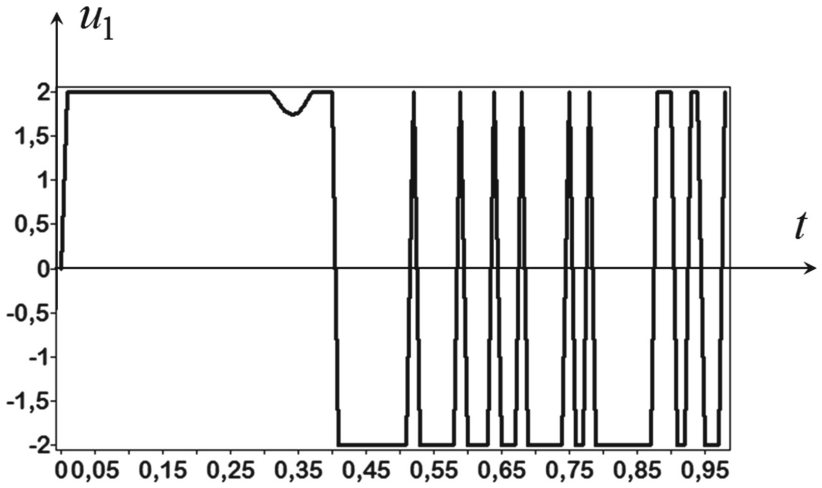


Fig. 4. Control u_1 for initial condition $\mathbf{x}^{0,1}$

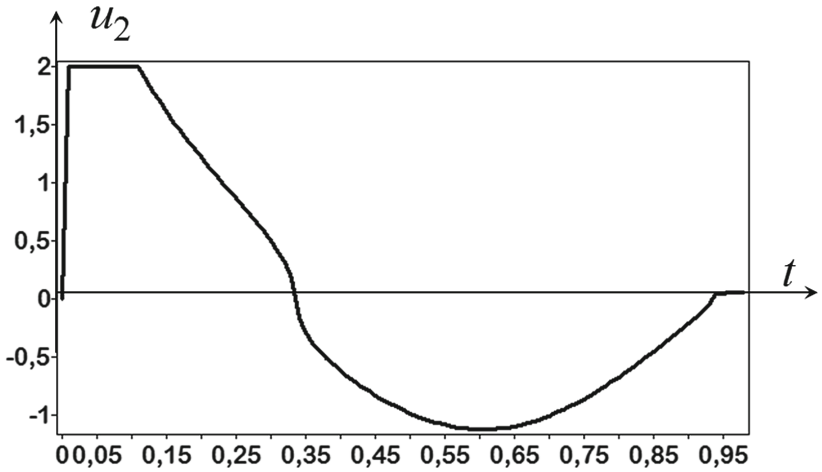


Fig. 5. Control u_2 for initial condition $\mathbf{x}^{0,1}$

$$
\begin{aligned}
\dot{x}_1 &= x_4 + (x_5 \sin(x_1) + x_6 \cos(x_1)) \sin(x_2)/cos(x_2),\\
\dot{x}_2 &= (x_5 \sin(x_1) + x_6 \cos(x_1))/\cos(x_2),\\
\dot{x}_3 &= x_5 \cos(x_1) + x_6 \sin(x_1),\\
\dot{x}_4 &= x_5 x_6 (I_2 - I_3)/I_1 + \tilde{h}_1(\Delta\mathbf{x})/I_1,\\
\dot{x}_5 &= x_4 x_6 (I_3 - I_1)/I_2 + \tilde{h}_2(\Delta\mathbf{x})/I_2,\\
\dot{x}_6 &= x_4 x_5 (I_1 - I_2)/I_3 + \tilde{h}_3(\Delta\mathbf{x})/I_3,
\end{aligned}
\tag{33}
$$

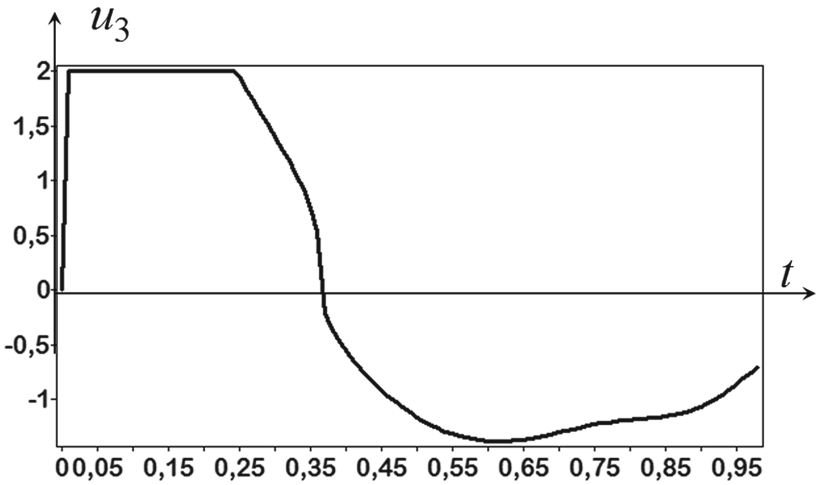


Fig. 6. Control u_3 for initial condition $\mathbf{x}^{0,1}$

where

$$\tilde{h}_i(\Delta\mathbf{x}) = u_i,\ i = 1, 2, 3, \tag{34}$$

$$\Delta\mathbf{x} = \begin{bmatrix} x_1^* - x_1 \\ x_2^* - x_2 \\ x_3^* - x_3 \\ -x_4 \\ -x_5 \\ -x_6 \end{bmatrix}. \tag{35}$$

Now the optimal control system synthesis problem is solved for stabilization of object in a point of six-measure space $\{x_7, \ldots, x_{12}\}$. The control of object include four components

$$\hat{\mathbf{u}} = [x_1^*\ x_2^*\ x_3^*\ u_4]^T. \tag{36}$$

In this problem the point $\mathbf{x}^* = [x_7^* \ldots {}_{12}^*]^T$ in six-measure space is set and the vector of control (36) is searched for the systems (25), (33). To solve this problem the network operator method is applied too. In the problem a set of initial condition included eight elements.

$$\begin{aligned} \mathrm{X}_0 = \{&\mathbf{x}^{0,1} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ -0.5\ -0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,2} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ -0.5\ 0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,3} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ 0.5\ -0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,4} = [0\ 0\ 0\ 0\ 0\ 0\ -0.5\ 0.5\ 0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,5} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ -0.5\ -0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,6} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ -0.5\ 0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,7} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0.5\ -0.5\ 0\ 0\ 0]^T, \\ &\mathbf{x}^{0,8} = [0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0.5\ 0.5\ 0\ 0\ 0]^T\}. \end{aligned} \tag{37}$$

Constraints on control were

$$\begin{array}{r} -pi/4 = x_1^- \le x_1^* \le x_1^+ = \pi/4, \\ -pi/4 = x_2^- \le x_2^* \le x_2^+ = \pi/4, \\ -pi/4 = x_3^- \le x_3^* \le x_3^+ = \pi/4, \\ 0 = u_4^- \le u_4 \le u_4^+ = 12. \end{array} \tag{38}$$

Terminal condition was

$$\mathbf{x}^f = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 2\ 0\ 0\ 0\ 0]^T. \tag{39}$$

The next functional is used

$$J = \sum_{i=1}^{8} \left(t_{f,i} + a_2 \sqrt{\sum_{j=1}^{12} (x_j^f - x_j(t_{f,i}, \mathbf{x}^{0,i}))} \right), \tag{40}$$

where a_2 is a weight coefficient, $a_2 = 2.5$, $t_{f,i}$ is a terminal time for solution with initial condition $\mathbf{x}^{0,i}$, a terminal time is defined by Eq. (5) with $\varepsilon_1 = 0.05$, $t^+ = 2$ $\mathbf{x}(t, \mathbf{x}^{0,i})$ is a partial solution of the system (24) with control (26) and initial conditions $\mathbf{x}^{0,i}$, $i \in \{1, \ldots, 8\}$.

The parameters of algorithm where: a number of possible solution in the initial set $H = 1024$, the number of generations $P = 128$, the number of crossovers in one generation $R = 128$, a dimension of the network operator matrix 32×32, the number of variation vectors in one possible solution $l_1 = 6$, the number of generations between exchange of basic solution 10, probability of mutation $p_\mu = 0.7$.

A basic solution was

$$\begin{array}{l} x_1^* = q_{11}(x_9^f - x_9) - q_{12} \\ x_2^* = q_7(x_7^f - x_7) - q_8 x_{10} + q_9(x_8^f - x_8) - q_{10} x_{11} \\ \qquad + q_{11}(x_9^f - x_9) - q_{12} x_{12}, \\ x_3^* = q_7(x_7^f - x_7) - q_8 x_{10}, \\ u_4 = q_9(x_8^f - x_8) - q_{10} x_{11}. \end{array} \tag{41}$$

As the result the following solution was received

$$x_i^* = \begin{cases} x_i^+, \text{ if } \tilde{x}^i > x_i^+ \\ x_i^-, \text{ if } \tilde{x}^i < x_i^- \\ \tilde{x}_i^* - \text{ otherwise} \end{cases}, \ i = 1, 2, 3 \tag{42}$$

$$u_4 = \begin{cases} u_4^+, \text{ if } \tilde{u}_4 > u_4^+ \\ u_4^-, \text{ if } \tilde{u}_4 < u_4^- \\ \tilde{u}_4 - \text{ otherwise} \end{cases}, \tag{43}$$

$$\tilde{x}_1^* = (A_2 q_9(x_9^f - x_9)\cos(x_{11})\exp(-q_{12}))\sqrt[3]{A_2}\log(|q_9(x_9^f - x_9)\cos(x_{11})|), \tag{44}$$

$$\begin{aligned}\tilde{x}_2^* = {} & \sqrt[3]{\tilde{x}_1^*} + 2\arctan(C_2) - q_{10}^3 x_{10}^9 + D_2 - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 + \\ & -q_7(x_7^f - x_7) + \arctan(A_2 q_9(x_9^f - x_9)\cos(x_{11})\exp(-q_{12})) \\ & + \arctan(-q_{10}x_{10}^3 + q_7(x_7^f - x_7)) + q_8(x_8^f - x_8) \\ & + \operatorname{sgn}(-q_{10}x_{10}^3 + q_7(x_7^f - x_7))\sqrt{|-q_{10}x_{10}^3 + q_7(x_7^f - x_7)|} \\ & + \tanh(-0.5x_{12}) + \mu(A_2 q_9(x_9^f - x_9)\cos(x_{11})\exp(-q_{12})) \\ & + \operatorname{sgn}(E_2)(\exp(|E_2|) - 1),\end{aligned} \tag{45}$$

$$\begin{aligned}\tilde{x}_3^* = {} & \operatorname{sgn}(\tilde{x}_1^*)\log(|\tilde{x}_1^*| + 1) + \sqrt[3]{B_2} + \arctan(C_2) - q_{10}^3 x_{10}^9 \\ & - \tilde{x}_1^* + (-q_{10}x_{10}^3 + q_7(x_7^f - x_7))^3\end{aligned} \tag{46}$$

$$\begin{aligned}\tilde{u}_4 = {} & \sin(\tilde{x}_3^*) + \operatorname{sgn}(\tilde{x}_2^*)(\exp(|\tilde{x}_2^*|) - 1) + \operatorname{sgn}(\tilde{x}_1^*)\log(|\tilde{x}_1^*| + 1) \\ & + C_2^2 + (A_2 q_9(x_9^f - x_9)\cos(x_{11})\exp(-q_{12}))^2 + \tanh(0.5B_2) \\ & + \tanh(0.5E_2) + (-q_{10}x_{10}^3 + q_7(x_7^f - x_7))^2,\end{aligned} \tag{47}$$

where

$$A_2 = q_{12} - x_{12} + \sqrt[3]{q_{10}} + \arctan(q_9) + \cos(x_7^f - x_7),$$

$$B_2 = \arctan(-q_{10}x_{10}^3 + q_7(x_7^f - x_7)) - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 + q_8(x_8^f - x_8) - q_7(x_7^f - x_7),$$

$$\begin{aligned}C_2 = {} & \operatorname{sgn}(-q_{10}x_{10}^3 + q_7(x_7^f - x_7))(exp(|-q_{10}x_{10}^3 + q_7(x_7^f - x_7)|) - 1) \\ & + \operatorname{sgn}(-q_{10}x_{10}^3)\sqrt{|-q_{10}x_{10}^3|} + \mu(q_7(x_7^f - x_7)) + q_7^3,\end{aligned}$$

$$\begin{aligned}D_2 = {} & 2(\arctan(-q_{10}x_{10}^3 + q_7(x_7^f - x_7)) - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 + q_8(x_8^f - x_8) \\ & + \operatorname{sgn}(-q_{11}x_{11}q_8^2(x_8^f - x_8)^2)\exp(-|-q_{11}x_{11}q_8^2(x_8^f - x_8)^2|) + \exp(q_{10}) - 1 \\ & + (-q_{10}x_{10}^3 + q_7(x_7^f - x_7))^3 - q_7(x_7^f - x_7))+,\end{aligned}$$

$$E_2 = \arctan(-q_{10}x_{10}^3 + q_7(x_7^f - x_7)) - q_{11}x_{11}q_8^2(x_8^f - x_8)^2 + q_8(x_8^f - x_8) - q_7(x_7^f - x_7),$$

$$\mu(\beta) = \begin{cases}\operatorname{sgn}(\beta), \text{ if } |\beta| > 1 \\ \beta\text{– otherwise}\end{cases}.$$

$q_7 = 0.115$, $q_8 = 3.371$, $q_9 = 3,076$, $q_{10} = 0,144$, $q_{11} = 3.131$, $q_{12} = 4.515$.

At the search process, the criterion (40) was counted 1,189,440 times. Calculation time was 1.5 h.

In Fig. 7 and 8, the helicopter trajectories of movement from eight initial conditions are shown.

In Fig. 9, 10, 11 and 12 plots of control in spatial spaces are shown.

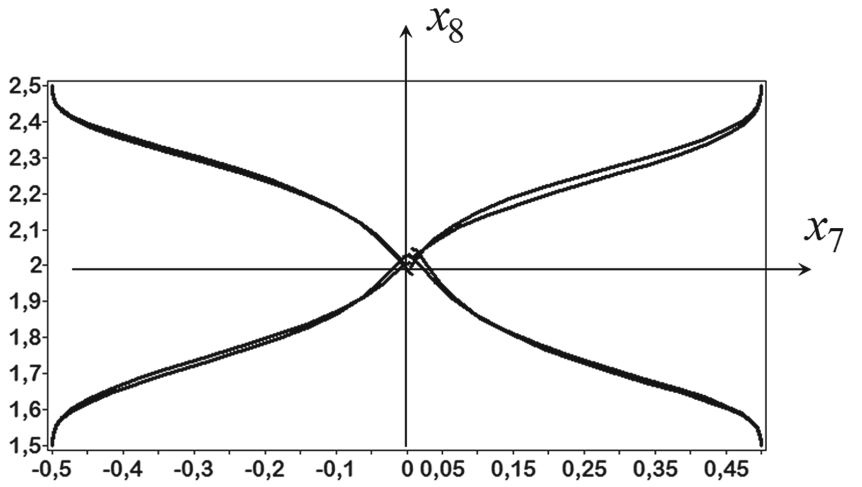


Fig. 7. Trajectories of helicopter in vertical plane $\{x_7, x_8\}$ from eight initial conditions

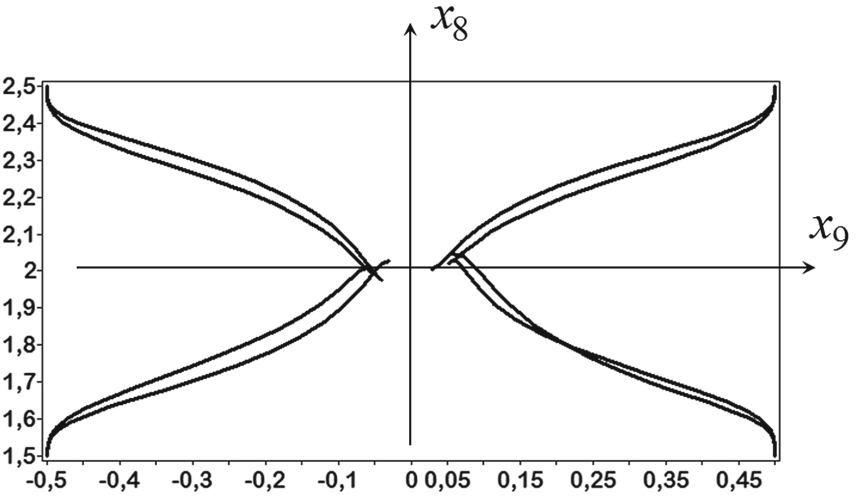


Fig. 8. Trajectories of helicopter in vertical plane $\{x_9, x_8\}$ from eight initial conditions

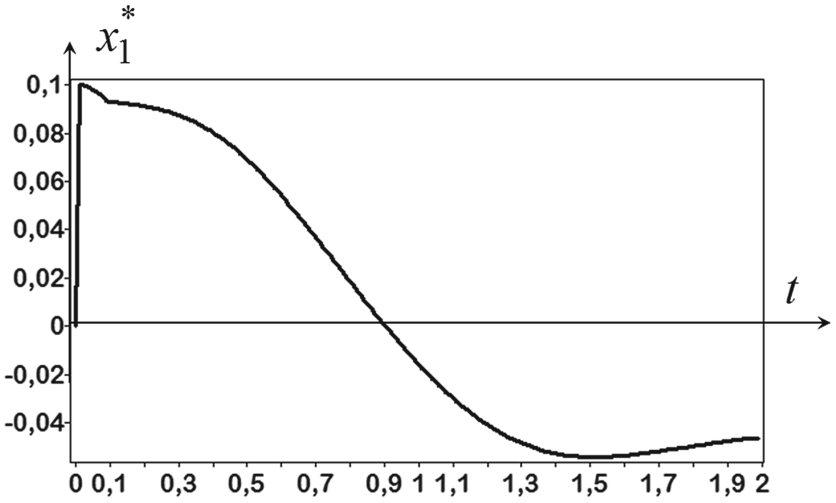


Fig. 9. Control $\tilde{x}_1^*$

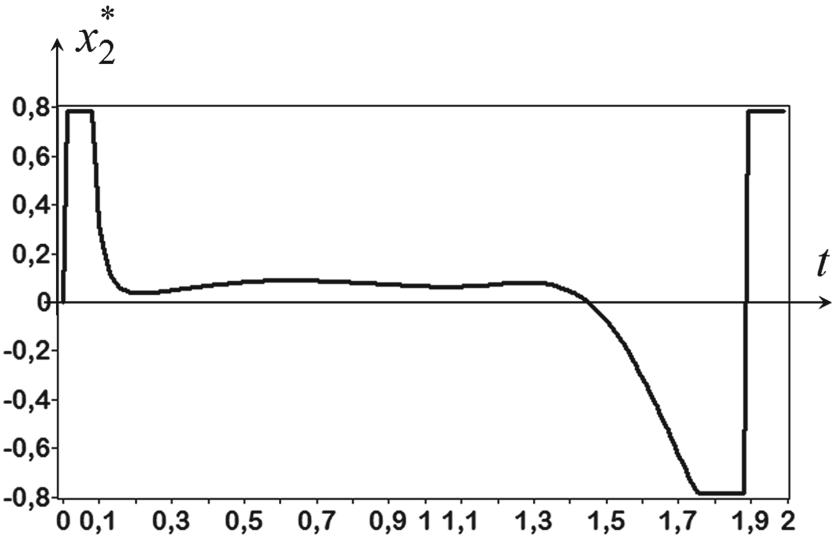


Fig. 10. Control $\tilde{x}_2^*$

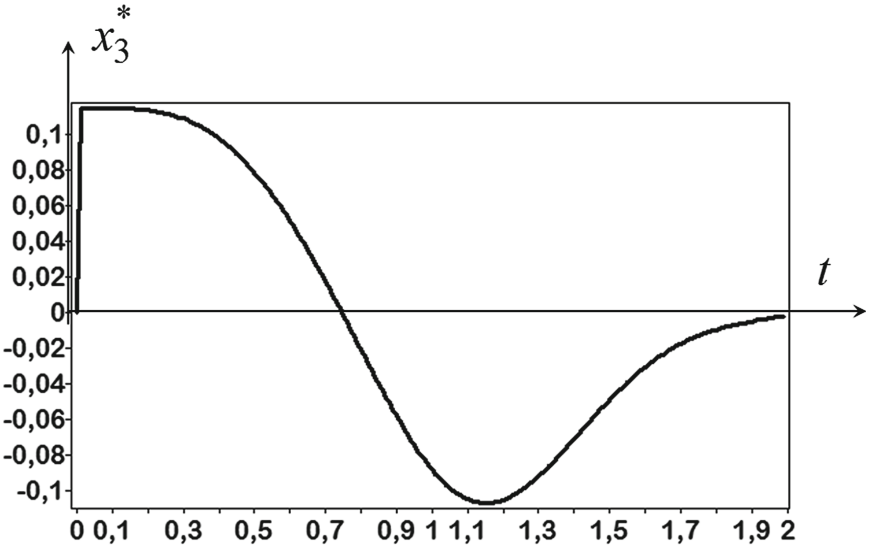


Fig. 11. Control $\tilde{x}_3^*$

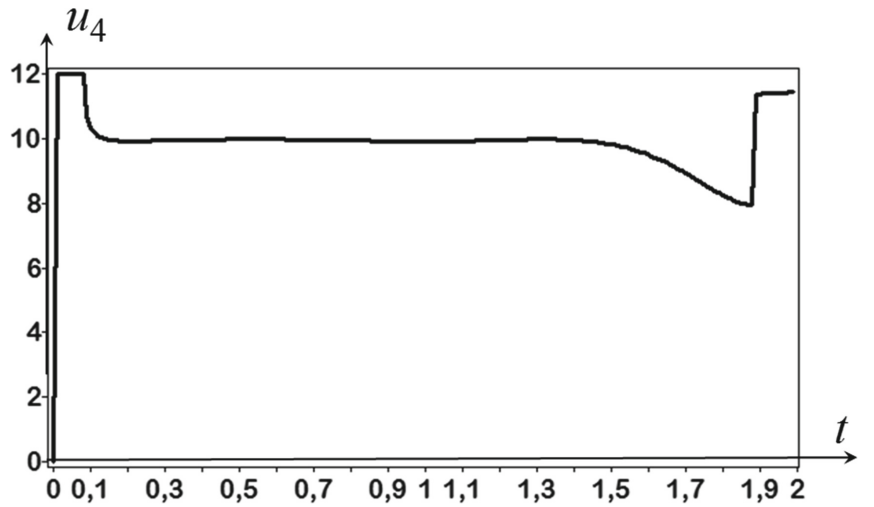


Fig. 12. Control $\tilde{u}_4$

4 Conclusion

The paper discusses the use of the numerical evolutionary method of symbolic regression to solve the problem of control system synthesis. A description of one of the symbolic regression methods is provided. It is explained that symbolic regression methods encode possible solutions in a convenient form for computer

processing and search for the optimal solution using a genetic algorithm in the space of codes. It is shown that the application of the symbolic regression method allows to automate the process of solving the complex problem of the control system synthesis without carrying out analytical transformations. As an example, a problem of synthesizing a quad-rotor helicopter control system is considered. The problem was carried out in two stages. At the first stage, the synthesis problem of the angular stabilization system was solved. At the second stage the problem of spatial stabilization was solved due to control of lift and angles of quadrocopter inclination. Simulation showed the efficiency of the synthesized control system.

Acknowledgment. The work, sections was performed at support from the Russian Science Foundation (project No 19-11-00258).

References

1. Clarke, F.: Lyapunov functions and feedback in nonlinear control. In: de Queiroz, M., et al. (eds.) Optimal Control, Stabilization and Nonsmooth Analysis. LNCIS, vol. 301, pp. 267–282. Springer, Heidelberg (2004)
2. Vinter, R.: Convex duality and nonlinear optimal control. SIAM J. Control Optim. **31**(2), 518–538 (1993). https://doi.org/10.1137/0331024
3. Mizhidon, A.D.: On a problem of analytic design of an optimal controller. Autom. Remote Control **72**(11), 2315–2327 (2011). https://doi.org/10.1134/S0005117911110063
4. Podvalny, S.L., Vasiljev, E.M.: Analytical synthesis of aggregated regulators for unmanned aerial vehicles. J. Math. Sci. **239**(2), 135–145 (2019). https://doi.org/10.1007/s10958-019-04295-w
5. Diveev, A.I.: A numerical method for network operator for synthesis of a control system with uncertain initial values. J. Comput. Syst. Sci. Int. **51**(2), 228–243 (2012). https://doi.org/10.1134/S1064230712010066
6. Diveev, A.I.: Small variations of basic solution method for non-numerical optimization. IFAC-PapersOnLine **48**(25), 028–033 (2015). https://doi.org/10.1016/j.ifacol.2015.11.054
7. Diveev, A.I., Sofronova, E.A.: Automation of synthesized optimal control problem solution mobile robot by genetic programming. In: Bi, Y., et al. (eds.). Intelligent Systems and Applications. Advances in Intelligent Systems and Computing, Vol. 1038. Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), 5–6 September 2019, London, UK, vol. 2, pp. 1054–1072. Springer, Heidelberg (2020).https://doi.org/10.1007/978-3-030-29513-4_77
8. Diveev, A.I., Sofronova, E.A.: The network operator method for search of the most suitable mathematical expression. In: Gao, S. (ed.) Bio-inspired Computational Algorithms and Their Applications, pp. 19–42. Intech (2012). (Chapter 2, ISBN 978-953-51-0214-4)

Navigation Stack for Robots Working in Steep Slope Vineyard

Luís C. Santos[1,2(✉)], André S. Aguiar[1,2], Filipe N. Santos[1], António Valente[1,2], José Boa Ventura[1,2], and Armando J. Sousa[1,3]

[1] INESC TEC - Institute for Systems and Computer Engineering, Technology and Science, Porto, Portugal
{luis.c.santos,andre.s.aguiar,fbsantos}@inesctec.pt
[2] UTAD - University of Trás-os-Montes e Alto Douro, Vila Real, Portugal
{avalente,jboavent}@utad.pt
[3] FEUP - Faculty of Engineering of University of Porto, Porto, Portugal
asousa@fe.up.pt

Abstract. Agricultural robotics is nowadays a complex, challenging, and relevant research topic for the sustainability of our society. Some agricultural environments present harsh conditions to robotics operability. In the case of steep-slope vineyards, there are several robotic challenges: terrain irregularities, characteristics of illumination, and inaccuracy/unavailability of the Global Navigation Satellite System. Under these conditions, robotics navigation, mapping, and localization become a challenging task. Performing these tasks with safety and accuracy, a reliable and advanced Navigation stack for robots working in a steep slope vineyard is required. This paper presents the integration of several robotic components, path planning aware of robot centre of gravity and terrain slope, occupation grid map extraction from satellite images, a localization and mapping procedure based on high-level visual features reliable under GNSS signals blockage/missing, for steep-slope robots.

Keywords: Autonomous navigation · Localization · Mapping · Path planning · Agricultural robotics

1 Introduction

Autonomous robot navigation aggregates different research areas in robotics such as localization, mapping, path planning, motion control and decision making [23]. The steep slope vineyards placed in Douro Demarcated Region, UNESCO Heritage place, present unique characteristics, Fig. 1, which includes complex topographic and soil profiles. This generates various robotic challenges to reach a fully autonomous navigation system. The terrain characteristics produce signal blockage, which decreases the availability and accuracy of Global Navigation Satellite Systems (GNSS); the harsh terrain condition affects the accuracy of dead-reckoning sensors like odometry or inertial measurement systems (IMU);

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 264–285, 2021.
https://doi.org/10.1007/978-3-030-55180-3_21

Fig. 1. Typical Douro's steep slope vineyard

the terrain strong slopes imposes constrains to the robot path planning. Safe navigation requires an accurate map of the vineyard and precise information about its posture (localization and attitude) [8].

This works aims to present a stack for autonomous robot navigation on steep slope vineyards. The stack is composed of a GNSS-free localization and mapping approach (VineSlam), and an advanced path planning framework for navigation in steep slope vineyards (Agricultural Robotic Path Planning - AgRobPP). The localization approach considers the detection of natural vineyard features (trunks and masts) in order to allow a hybrid Simultaneous Localization and Mapping (SLAM), based on topological and features maps increase the localization accuracy and robustness [7]. The path planning method is a framework which allows the robot to travel from one location to another, avoiding obstacles and dangerous slopes that would present risks to the robotic platform or centenary vine-trees. The proposed path planning method is an extended cell decomposition planner with A* algorithm aware of the robot's orientation and centre of mass. The framework requires an occupation grid map and a digital elevation model (DEM), to reach a safe path. Besides, AgRobPP provides a tool to extract these maps from a 3D point cloud obtained by any agricultural robot [30]. This framework includes different extensions to consider different task scenarios, for example, avoid soil compaction and spraying tasks.

Section 2 presents the related work in the literature about localization and path planning. Section 3.1 presents our robotic platform, AgRobv16. Section 4 provides the approach for the natural feature detection important for the implementation of VineSlam. Section 5 presents the approach used for path planning, AgRoBPP, as well as the extensions of this framework. Besides, this section presents a tool to extract a draft map from high-resolution satellite images. Section 6 introduces a system for robotic software safety verification. Section 7 presents the results of the localization and path planning system, and Sect. 8 contains the conclusions and future work topics.

2 Related Work

The autonomous robot navigation is a widely explored issue due to its importance in the design of intelligent mobile robots. The Localisation, Mapping and Path planning processes are crucial for autonomous tasks. Depending on the robotic application, a solution for SLAM is frequently necessary. The SLAM problem for indoor scenarios is widely explored with a considerable number of approaches available in the literature. Bailey *et al.* [3] considered the Extended Kalman Filter (EKF) to be the most used method for SLAM at the time. The system state vector of EKF-based SLAM contains the pose of the vehicle and parameters which describe the environment features. There are several variants based on feature maps such as Unscented Kalman Filter (UKF) SLAM, EKF-based FastSLAM [15]. Unlike indoor scenarios, agriculture scenarios expose the robot to non-structured environments, with adverse conditions. This affects the extraction of features for SLAM. Faessler *et al.* [10] uses images key-points to estimate visual odometry; however, in agricultural fields, the key points are associated with leaves and grass which are highly affected by the wind and other atmospheric conditions. Cheein *et al.* [4] presents a SLAM approach based on the detection of olive stems, detected by a vision system with a support vector machine (SVM). Outliers on the detection and wrong closed-loop association were the main drawbacks in this approach. Cheein *et al.* [5] presents a survey on features detection in agriculture focused on orchards and vegetables farms. However, vineyards different natural features. Identification of natural features in vineyards is essential to construct a hybrid SLAM approach independent of GNSS, to create a robust localisation system. Other redundant localisation systems based in artificial beacons have been studied, like in the work of Hahnel *et al.* [13], that estimate the localisation of artificial RFID beacons. Pinto *et al.* [21] presents a sensor fusion-based approach using Bluetooth beacons.

The path planning task is essential for robotic navigation and consists of the capability to find a path between a start point and a destination point while avoiding obstacles and optimising parameters like time, distance or energy. The path planning algorithms are composed of two main categories: “off-line” and “on-line”. “Off-line” path planning is used when the robot has previous access to the complete information about the environment, including the trajectories of moving obstacles. When this information is incomplete or nonexistent, the robot completes it during navigation, being this method known as “On-line” path planning [23]. The path planning methods consist of various concepts such as potential field, sampling-based method and cell decomposition. In potential field planners, the robot behaves as a particle immersed in a potential field, where the goal point represents an attraction potential, and the obstacles represent repulsive potentials. This algorithm is not optimal; that is, it does not always generate the best possible path according to the requested requirements. Besides, the local minimum problem might stop the robot from reaching the destination [22]. Rapidly exploring random tree (RRT) is a sampling-base method, which explores the path randomly. Even though these planners are simple, they are not optimal and tend to generate paths with abrupt changes of direction

[25]. However, Karaman *et al.* [14] developed a motion planning system with RRT*, a modified version that converges to a near-optimal method. In the cell decomposition method, the free space is divided into small regions called cells, where each cell contains information about its availability (existence of obstacle or free space). Considering these maps, search algorithms like A* or Dijkstra can search for a path between two points. A* is a variant of Dijkstra that uses a heuristic to find the optimal path between two points. Cell decomposition with A* is an optimal algorithm but has high computational complexity. There are several variations of the original A* algorithm to improve certain characteristics, such as processing time [12], dynamic obstacles [19], and robot constraints [11]. It is required to construct an adequate path planning strategy that allows the robot to safely navigate in a steep slope vineyard considering all of the terrain irregularities, obstacles and potential signal localisation failures. To the best of our knowledge, there is no solution that fulfills the presented requirements for path planning and localisation, required for steep slope vineyards scenarios. It is required that the robot should be capable of navigating safely, independent of GNSS systems, taking into account the terrain irregularities, obstacles and potential signal localisation failures.

3 Navigation Stack

The proposed navigation stack contains two main components: Localization and Mapping, and Path Planning. Figure 2 illustrates the high-level architecture of the navigation stack which requires a hybrid map composed by a topological map, a feature-based map, and a 3D map. The localization and mapping stack uses different sensors to perform high-level feature detection, that is used to construct a feature-based map. The 3D map is constructed with a registry of sensors measurements along the time. The path planning stack initiates with a pre-knowledge map obtained from satellite image segmentation (if available), which can be used to obtain a 2D occupancy map. The tool extracts both Occupancy and elevation maps from a 3D map, in order to complete the pre-knowledge map. Besides, from the occupancy map, a topological map is generated to improve path planning efficiency and auxiliary the localization system. The robotic platform considered in this research work is described in the section below.

3.1 AgRobv16

AgRobv16 is a robotic platform for research and development of robotics technologies for Douro's vineyards. The first version of this platform, Fig. 3(a), is equipped with an IMU, Odometry sensors, low-cost GNSS sensor, 2D Light Detection And Ranging Sensors (LIDARs), stereo camera and one monocular camera with special light filters to extract Normalized Difference Vegetation Index (NDVI) images. The second version, Fig. 3(b), contains a different set of sensors, composed by IMU, low-cost GNSS sensor, 2D LIDAR, 3D LIDAR (Velodyne VLP-16), stereo camera and thermal camera. The platform is also

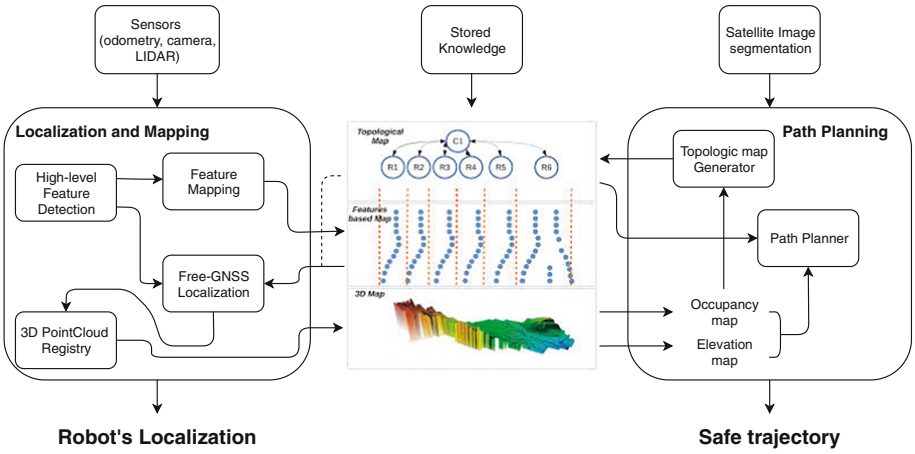


Fig. 2. The proposed high-level architecture of the navigation stack

(a) (b)

Fig. 3. AgRobv16 robotic Platform (a) - AgRobv16 v1; (b) - AgRobv16 v2

equipped with a robotic arm manipulator for future for monitoring or pruning tasks.

4 Localization and Mapping

The localization system presented in the proposed navigation stack considers three fundamental steps: feature detection, feature mapping, and localization considering the generated map. The system is composed of several approaches, aiming the same goal, that is here briefly described.

4.1 Feature Detection

The proposed navigation stack considers high-level features extracted from the vineyard, the vine masts, and trunks. In order to perform a reliable detection of the trunks, we propose the Vine Trunk Detector - ViTruDe [16,17]. ViTruDe is composed of three main components: key-points extractor, key-points search and region descriptor extractor, and Support Vector Machine (SVM) classification. Figure 4 represents the ViTruDe approach.

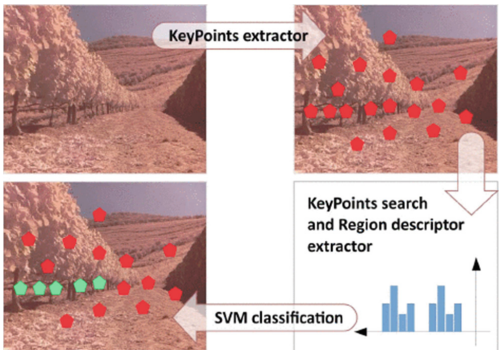


Fig. 4. Information flow and main components of ViTruDe [16]

The method first detects key-points in the input image using the Sobel operator, where a small region is analyzed for the presence of masts, and trunks. Then, ViTruDe extracts a descriptor in the region around each key-point, that constitute the input for the SVM classifier. The descriptor is computed using the Local Binary Pattern (LBP) [20] code. LBP is a sturdy grey-level invariant texture primitive, used for textured image description. To describe the image texture, a LBP histogram (hLBP) is built from all binary patterns of each image pixel, represented in Eq. 1:

$$\begin{cases} H(k) = \sum_{m=1}^{M} \sum_{n=1}^{N} f(LBP_{P,R}(m,n),\ k), \quad k \in [0,K] \\ f(x,y) = \begin{cases} 1, & x = y \\ 0, & otherwise \end{cases} \end{cases} \tag{1}$$

where K is the maximal LBP pattern value.

Based on the training step, the SVM is able to classify if a mast or trunk is present in that small image window.

Recently, a new approach has been developed based on Convolutional Neural Networks (CNN). A dataset containing a set of training vine images, with manually annotated trunk locations that are publicly available (http://vcriis01.inesctec.pt/, with the `DS_AG_39` id) was used to train several CNNs. The training procedure considers pre-trained models, retraining them, using a fine-tuning approach.

4.2 Feature Mapping

After having a reliable high-level feature detector, a global map of the masts and trunks can be computed. The fundamental mapping approach of the proposed navigation stack is implemented on VineSlam [8]. This work builds a hybrid map - topological, features, and metric-map - to deal with natural and artificial landmarks.

The proposed approach considers artificial landmarks placed in the beginning and end of each vine row, namely, Radio-Frequency IDentification (RFID) tags [9]. The hybrid map architecture builds two different maps, one topological, and other useful features. In the first, each node symbolizes a vineyard row, and an RFID-based landmark detection represents the transitions between nodes. Each node stores a metric feature-based map, RFID tags associated with the row, and altitude range of the row. The information of altitude range increases the localization robustness, in the case of steep slope vineyards.

Using the recent landmark detection approach based in CNN models, a new feature mapping technique is also proposed in this stack. This is based in the EKF proposed in FastSLAM [18], and a single EKF is considered for each detected landmark. Each EKF is initialized when a sufficient set of observations k is achieved. For each landmark j at a given time i there are two unknowns: the trunk j distance to the robot at the time i λ_{ij} and the global position of the trunk s_j. Using the odometry estimation for the robot pose $[m_{xi}, m_{yi}, m_{\theta i}]$ and the bearing observation ψ_{ij} we have

$$\begin{cases} s_{xj} = m_{xi} + \lambda_{ij} cos(m_{\theta i} + \psi_{ij}) \\ s_{yj} = m_{yi} + \lambda_{ij} sin(m_{\theta i} + \psi_{ij}) \end{cases} \tag{2}$$

Considering the k observations, this equation can be rearranged, as proposed in [6], in the following form

$$\begin{bmatrix} 1 & 0 & -cos(\phi_{1j}) & 0 & \dots & 0 \\ 0 & 1 & -sin(\phi_{1j}) & 0 & \dots & 0 \\ 1 & 0 & 0 & -cos(\phi_{2j}) & \dots & 0 \\ 0 & 1 & 0 & -sin(\phi_{2j}) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & 0 & \dots & -cos(\phi_{kj}) \\ 0 & 1 & 0 & 0 & \dots & -sin(\phi_{kj}) \end{bmatrix} \begin{bmatrix} s_{xj} \\ s_{yj} \\ \lambda_{1j} \\ \lambda_{2j} \\ \vdots \\ \lambda_{kj} \end{bmatrix} = \begin{bmatrix} m_{x1} \\ m_{y1} \\ m_{x2} \\ m_{y2} \\ \vdots \\ m_{xk} \\ m_{yk} \end{bmatrix} \tag{3}$$

where $\phi_{ij} = m_{\theta i} + \psi_{ij}$. If any of the depth parameters λ_{ij} results in a negative value, the initialization is discarded.

Since the landmark world position is intended to be static, the state model is as simple as

$$s_{j_t} = s_{j_{t-1}} \tag{4}$$

where s_{j_t} is the landmark position estimated at the instant t.

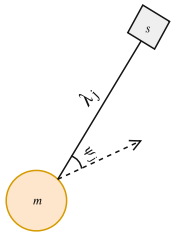


Fig. 5. Landmark observation configuration.

Considering the configuration present in Fig. 5, the observation model is implemented as described below. The measurement function $h(m_j, s_j) = [\lambda_j, \psi_j]^T$ is given by

$$h(m,s) = \begin{bmatrix} \lambda_j \\ \psi_j \end{bmatrix} = \begin{bmatrix} \sqrt{(s_{xj} - m_{xj})^2 + (s_{yj} - m_{yj}^2)} \\ atan(\frac{s_{yj} - m_{yj}}{s_{xj} - m_{xj}}) - m_{\theta j} \end{bmatrix} \tag{5}$$

The jacobian of h with respect to s is:

$$G_s = \begin{bmatrix} \frac{s_{xj} - m_{xj}}{\lambda_j} & \frac{s_{yj} - m_{yj}}{\lambda_j} \\ \frac{-(s_{yj} - m_{yj})}{\lambda_j^2} & \frac{s_{xj} - m_{xj}}{\lambda_j^2} \end{bmatrix} \tag{6}$$

Since with a monocular camera, only the landmark bearing is observable, its depth is calculated fusing wheel odometry with the orientation observation.

4.3 Localisation

VineSlam proposes a hybrid mapping approach, with three different layers: a topological map, a feature-based map, and a 3D map. In this work, a particle filter (PF) uses the feature-based map to localise the robot in real-time. The CNN-based mapping approach can also generate the map. Using an onboard camera on the robot, and given the previously built map of the vine, using the camera specifications is possible to determine which mapped vines are inside its field of view. These are reprojected back into the image, and matched with the online feature detector results. If a match is found, the particle representing a possible robot pose can be weighted using the distance from the detection and the mapped landmark.

In order to support the primary localisation approach, our system possesses several redundant localisation systems. The first is a redundant robot localisation system based in a wireless sensor network [24]. This work uses a multi-antenna receiver Bluetooth system, characterising the strength of the signal using the Received Signal Strength Indicator (RSSI). An EKF is used to compute the robot localisation using RSSI. Similarly, we also propose a localisation system based in ultra-wideband time-of-flight based technology (Pozyx). An EKF is used

to fuse wheel odometry information with Pozyx Range measurements. Additionally, a Visual Odometry (VO) localisation system called FAST-FUSION is also proposed [1,2].

This method uses a single fisheye camera to calculate the relative robot motion, fusing a gyroscope with it using a Kalman Filter (KF) to improve angular estimation, and a laser sensor to calculate the motion scale.

5 Path Planning

Before the initiation of any path planning process, a pre-knowledge occupation grid map is obtained through image analysis of satellite imagery. We resort to an SVM classifier to identify paths in public satellite images obtained from Google Maps [31]. Figure 6 shows two examples of the path classification. This approach will allow the robot to navigate autonomously under supervision, and construct a more detailed occupation map and elevation map, which will be explained in the sections below.

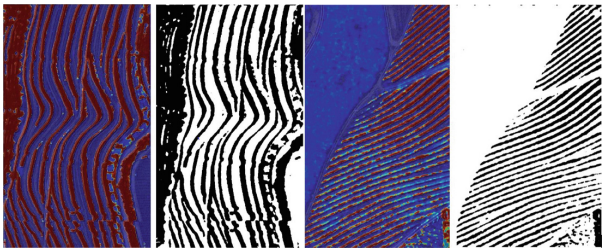


Fig. 6. Path classification in satellite image of steep slope vineyard. Black - Vegetation; White - Path; Coloured - Colour map of probability

To choose an adequate path planning algorithm, we considered parameters like intended optimization, computational complexity and method's efficiency. From literature, it stated that A* algorithm could be extended to improve specific parameters. In a previous work, Fernandes *et al.* [11] presented a modified A* algorithm that constraints the path to the maximum turn rate of the robot and optimizes safety region to rectangular shaped robots by considering the robot's orientation (yaw). Such feature is useful for navigation in the confined space of a vineyard. This approach originated the algorithmic approach AgRoBPP, which extends A* inner functions for path planning to be aware of the robot's centre of mass and terrain slope. Besides, this tool provides a method to extract an occupation map and an elevation map, called Point Cloud to Grid Map and DEM (PC2GD).

5.1 AgRobPP - Agricultural Robotics Path Planning

AgRobPP is the developed tool that contains the solution for safe path planning in steep slope vineyards. It is composed of:

- Update A* inner functions for the path planning to be aware of the robot's orientation and centre of mass;
- A method for safe local re-planning;
- PC2GD (Point Cloud to Grid Map and DEM);

A standard A* algorithm finds the best path with the following inputs: an occupation grid Map (cell decomposition method) and the robot's localization (x, y). Our approach also considers the centre of mass of the robot. Combining this information with a DEM, it is possible to generate a feasible and safe path for the robot. AgRoBPP extends the work of Fernandes *et al.* [11] which proposes an A* aware of the robot's orientation. This goal is reached by creating multiple grid-map layers which discretize the orientation into spaces of 22.5°; this defines the A* jump cells during the path search. The robot orientation discretization is made by the number of neighbours of a cell (square) equally radially spaced; this gives a number of $2^{(n+2)}$ neighbours, where n is the radius search space of a discretized circle. So, by dividing the circle into equal parts, it is possible to obtain a discretization of 45° (n = 1), 22.5° (n = 2) and 12.25° (n = 3) [30].

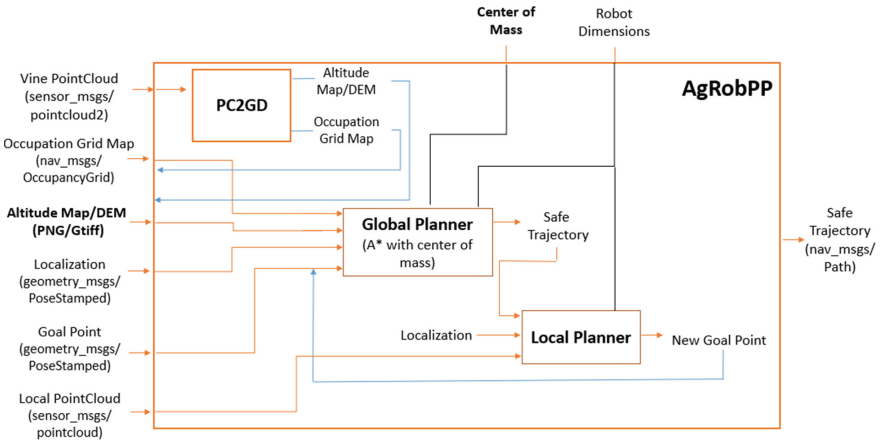


Fig. 7. AgRobPP diagram [30]

To reach the centre of mass constraint, two key inputs, Fig. 7, are considered: The Altitude Map/DEM and the robot's centre of mass provided by its coordinates (x, y, z) in the robot base referential. The generation of the safest path consists of the following steps:

1. Compute normal vectors to every single cell of the occupation grid map;
2. Check the imposed projection of the centre of mass in each cell taking into account the 16 layers of the map, that is, considering different robot's orientation;
3. Check the safety limit of the robot in each cell, blocking the dangerous cells.

The normal vectors are estimated the digital elevation map, which contains information about the coordinates (x, y, z) of each cell. With this information, it is possible to extract the roll and pitch of the cells and project the robot's centre of mass into the horizontal plane. Figure 8 shows the centre of mass projection for different robot's orientations.

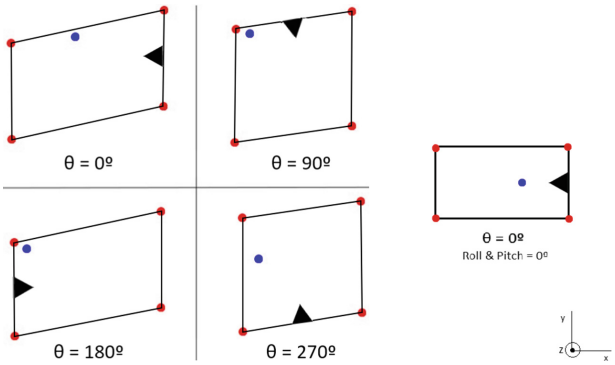


Fig. 8. Centre of mass projection in inclined zone (left) and on a plain zone (right). Red - Robot footprint; Blue - Center of mass projection; Triangle - front of the robot; θ - robot's orientation (yaw) [30]

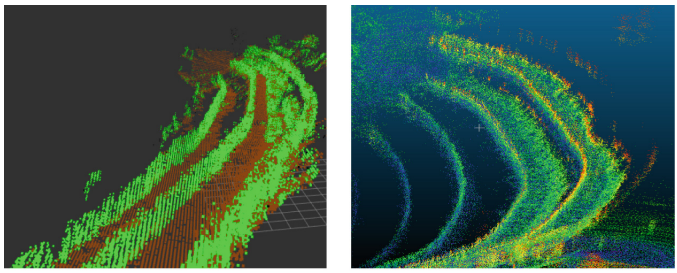
As it is impossible to guarantee a static environment, AgRobPP includes a local planner, Fig. 7. A 2D LIDAR placed in the front of the robot detects the presence of an obstacle and uses A* to generate a new path that will partially replace the previous trajectory. With this method, safety restrictions are considered.

5.2 Point Cloud to Grid Map and DEM - PC2GD

The previous stages can only be tested without this tool providing simulated data, that is, virtual maps of steep slope vineyards. In order to obtain real data, this method follows the next steps:

1. Extraction of tri-dimentional point cloud;
2. Point cloud segmentation;
3. Occupancy grid map extraction;
4. Extraction of elevation map;

The 3D map was, in a first stage obtained by a 2D LIDAR vertically positioned, with the segmented map visible in Fig. 9a. Later, these maps were obtained with a 3D LIDAR (VLP-16), Fig. 9b. These point clouds are useful to project a 2D occupation map and extract an elevation map.



(a) (b)

Fig. 9. (a) - Real vineyard Segmented PointCloud; Brown - floor; Green - vegetation/wall; (b) - Real vineyard PointCloud obtained with VLP-16

5.3 Extra Path Planning Extensions

According to the mission of the robot, path planning may depend upon several factors besides the initial and goal points. For different purposes, we have developed different extensions described below.

Path Planning for Automatic Recharging System. Automatic recharging methods are useful for long autonomous tasks. With these methods, the robot can perform long-time operations that would exceed the autonomy of its batteries. In the work of Santos *et al.* [29], we propose an off-line path planning approach to plan the trajectory to the nearest recharging point, and a docking method based on visual tags. The trajectories for the recharging points are previously generated according to energy costs, as a long downhill path may be more efficient than a shorter uphill path.

Path Planning Aware of Soil Compaction. Soil compaction is the process in which stress is applied to the soil, causing its densification as air is displaced from the pores between the soil grains. The intensive use of agricultural machinery is aggravating this problem, and agricultural robots may intensify soil compaction due to its capacity to replicate the same trajectories. To minimize this problem, in the work of Santos *et al.* [27], we have proposed an extension to the A* algorithm to avoid soil compaction. With the introduction of a compaction map, which recorded all the trajectories performed by the robot, the path planning algorithm can avoid repetitive paths, minimizing the soil compaction effects.

Path Planning Aware of Wall Vegetation (Vine Trees). Performing robotic agricultural tasks like monitoring, spraying, pruning or harvesting will require the robot to navigate at a certain distance from the vegetation wall (vine trees). For example, spraying tasks will require the robot to keep a distance to the vine trees to cover a certain area properly, and monitoring tasks may require to acquire a set of high-resolution pictures, which is highly affected by the distance. For this purpose, we developed an A* extension with a modified cost function to navigate preferentially at a desired distance from the vegetation wall [28].

Extraction of Vineyard Topological Maps. The big dimensions of agricultural terrains present a memory problem to the path planning task. The tool developed in the work of Santos *et al.* [28], proposes a solution resorting to topological maps. This tool identifies different zones of a vineyard grid map, with all possible transaction between these zones. With this, it is intended to prepare a path planning approach to work with hybrid maps, using topological and grid maps. Topological maps will be used to load only the necessary grid map zones of the map zones to generate a path.

6 Safety Verification for Robotic Software - SAFER

Modern robotic systems must be flexible, configurable and adaptive, requiring safety to be controlled by software, rather than through physical barriers or hardware mechanisms. SAFER project aims to develop such techniques for one of the most popular frameworks for developing robotic software, ROS. In this context, HAROS - High Assurance ROS [26], a static analysis tool of ROS software was considered to verify the quality of the developed code in AgRobPP. This tool verifies computational metrics and conformity with programming guides[1]. Table 1 shows some of the results of this tool's application.

Table 1. Static code analysis of AgRobPP code with HAROS

HAROS	Code metrics		Code style		Number of files	Total number of issues
	Excess of lines of code	Total	Non-specified integer	Total		
Before	34	75	744	5730	716	**5782**
After	17	41	32	2722	650	**2747**

7 Results

Many experiments using the navigation stack were performed in real-time on several steep slope vineyards. Here, a few results are presented for feature detection, feature mapping, localization and path planning.

[1] Google C++ Style Guide - https://google.github.io/styleguide/cppguide.html.

7.1 Feature Detection

Two approaches were proposed to perform feature detection. Figure 10 shows a result using the ViTruDe system.

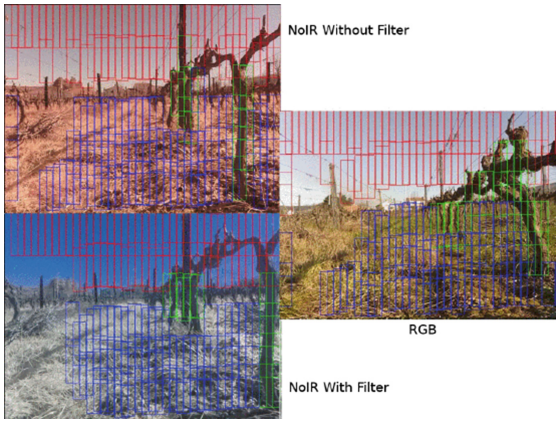
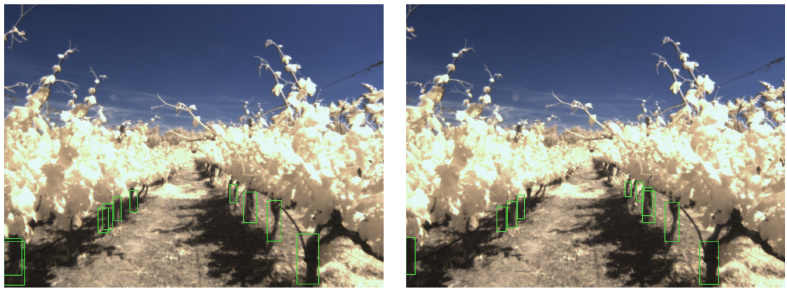


Fig. 10. Output results obtained by hLBP plus colour with four classes. Red rectangles classified as sky, green as mast/trunks, blue as ground [16].

Using an RGB and a NoIR camera, with and without filter, four classes are detected with success: sky, masts and trunks, and ground. Figure 11 shows an example of a processed image frame by the CNN proposed approach. It is visible that the DL-based approach presents a good performance extracting the trunks location on the images.



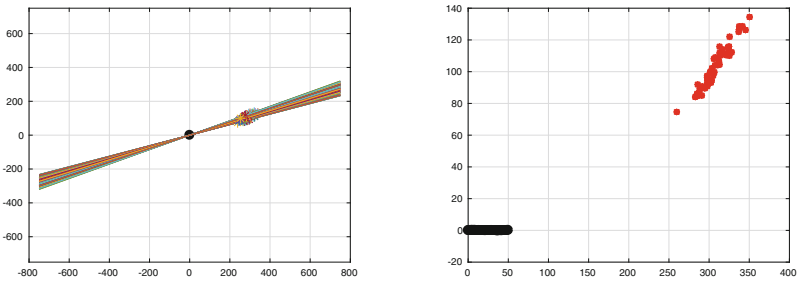
(a) (b)

Fig. 11. CNN-based trunk detection results.

7.2 Feature Mapping

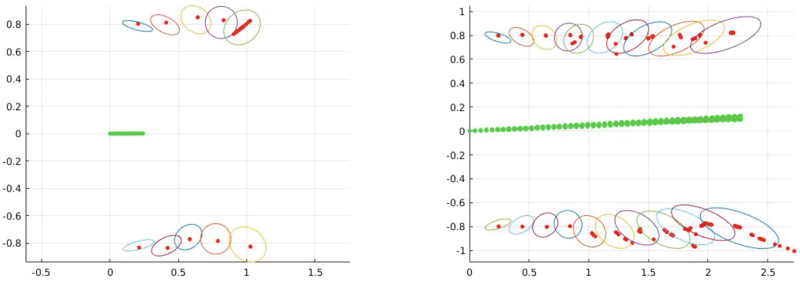
The vine map is constructed using the previously described EKF approach. A single trunk is mapped using the landmark bearing observation, and wheel odometry. Figure 12 shows a simulation of this procedure, for a single trunk. The bearing observation on the current image frame is projected in a line on the previous one, using the odometry measure. The two lines, corresponding to the two bearing observations, are intercepted, resulting in an estimation of the trunk position that is then used in the EKF. Figure 12(a) shows the interception of the lines, and Fig. 12b the final EKF estimation.

Figure 13 shows two simulated examples of two vine corridors mapping using the EKF approach.



(a) (b)

Fig. 12. EKF-based single trunk mapping (centimeters).



(a) (b)

Fig. 13. EKF-based vine corridor mapping (meters).

It is visible that the trunks are mapped with precision, and the error associated with the mapping is propagated during the robot motion.

7.3 Localization

Figure 14 overlaps the trajectory of the robot with the trajectory estimated by VineSLAM node. The orange line represents the real trajectory of the robot, and the blue line represents the estimated trajectory. During the vineyard row transition, the robot does not see any natural feature and estimates its localization with odometry information. So the estimation diverges from the real trajectory. On average, the VineSLAM localization error was 10.12 cm, with a standard deviation of 4.54 cm. So, with an accurate and robust natural feature detector, the VineSLAM can, in the future, accurately localize the robot using the proposed hybrid map [8].

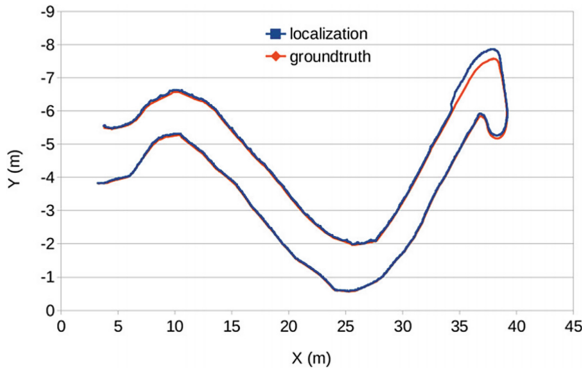


Fig. 14. Real robot trajectory in orange. Estimated robot trajectory in blue [8]

7.4 Path Planning

This section presents the results of AgRobPP with simulated and real data acquired by AgRobv16-v1 considering PC2GD. Rviz[2] tool was used to visualize the tests as well to select the start and destination points. The robot's dynamics was not simulated; only the generated paths were visualized to validate the proposed concept. The maps resolution is 5 cm/cell. Figure 15 represents a simulated occupation and elevation map of a steep slope vineyard, and Fig. 16 presents the application of PC2GD to the point cloud of Fig. 9, with an occupation grid map and an elevation map.

The generation of a safe path is illustrated in Fig. 17 an in Fig. 18, where it is compared to different variations of the A* algorithm without the centre of mass feature. Table 2 presents details about the robot posture in specific waypoints generated by the path planning algorithms. The results show that the A* without the centre of mass would impose risky posture on the robot.

[2] Rviz - http://wiki.ros.org/rviz.

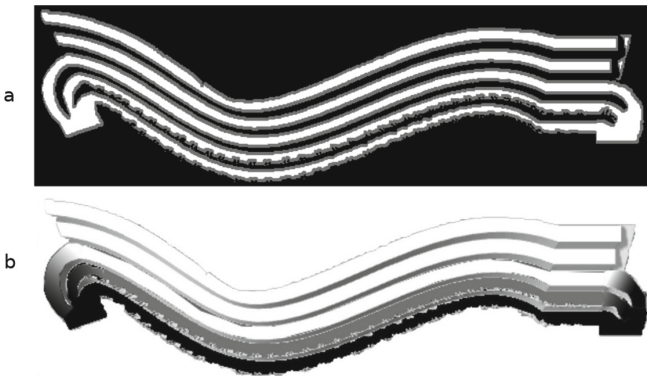


Fig. 15. Simulated maps of steep slope vineyard: a) Simulated occupation grid map; b) Simulated elevation map [30]

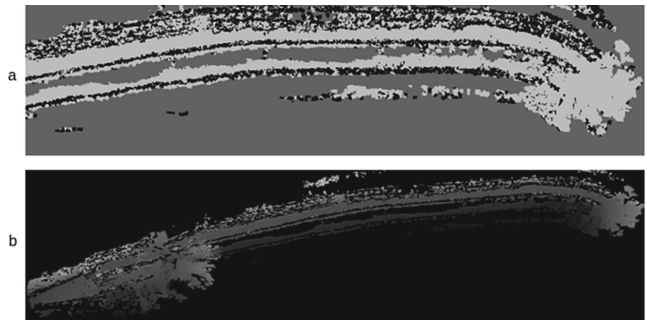


Fig. 16. PC2GD map extraction of real steep slope vineyard - a) Occupation grid map; b) Elevation map [30]

The results of the extra features developed for the path planning method are briefly expressed in Fig. 19. Figure 19 presents an example of a compaction map and the paths generated by different robot configuration after ten trajectories generated between the same two points avoiding soil compaction. Without this feature, all the paths would match the green one, which was generated without considering soil compaction. Figure 19b shows a topological representation of a vineyard grid-map. Figure 19c presents the paths generated with A* aware of vegetation wall, where the green path was generated by the regular A* while blue and orange paths were generated with this extension, considering 1 and 1.5 m of distance to the vine trees.

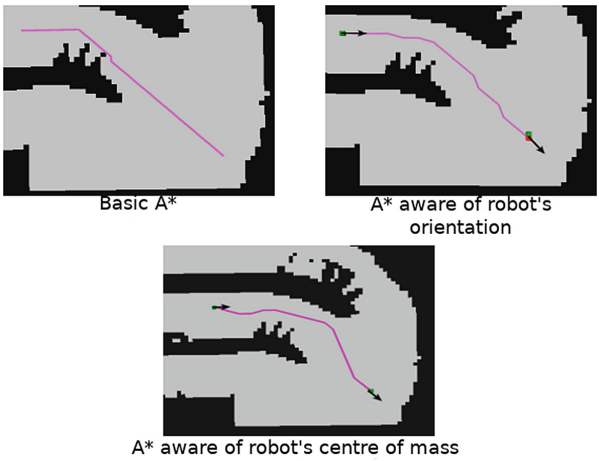


Fig. 17. Simulated data tests

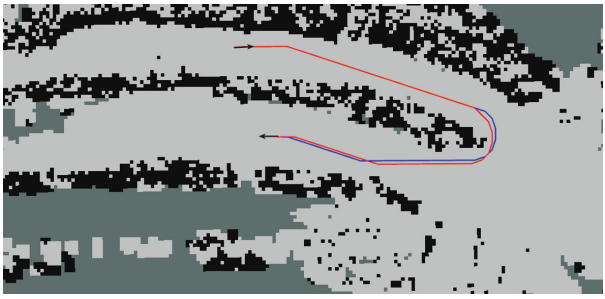


Fig. 18. Real data tests with A*: Red - A* with just orientation; Blue - A* aware of center of mass (AgrobPP)

Table 2. A* detailed results with real vineyard [30]

	Waypoint	Altitude (m)	Roll (degrees)	Pitch (degrees)	Yaw (degrees)	Safe zone?
A* with just orientation	6	−0.54	0.9	−0.9	0.0	Yes
	27	−0.7	45	45	90	No
	28	−0.8	45	45	112.5	No
AgRobPP	6	−0.54	0.9	0.9	90	Yes
	27	−0.66	0.9	0.9	90	Yes
	28	−0.78	0.0	2.3	112.5	Yes

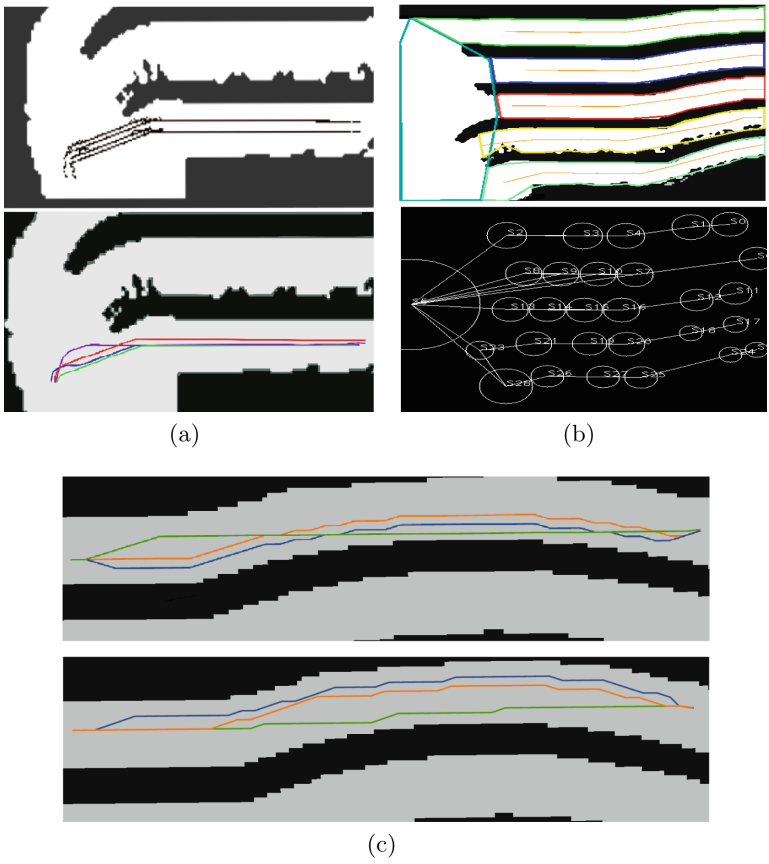


Fig. 19. A* extensions results - (a) A* avoiding Soil Compaction [27]; (b) Topological representation of grid map [28]; (c) A* aware of vegetation wall [28]

8 Conclusion

The proposed work presents an autonomous navigation stack for steep slope vineyards. This stack provides two main tools: Localization and Mapping, and Path Planning. The localization and mapping present VineSlam, a hybrid SLAM approach with topological and natural features. This approach comprises a natural feature detection tool, a feature mapping approach and a localization system based on natural and artificial features.

The path planning consists of an algorithmic approach, AgRobPP, with a cell decomposition planner using A*. This planner is aware of the robot's centre of mass and ensures the generation of a safe path for any robotic platform. Besides PC2GD tool, split a 3D PointCloud into two maps: a 2D occupation grid map and a high-resolution DEM. AgRobPP provides different extensions for specific tasks, like automatic recharging, avoid soil compaction and keep a distance to

the vine trees. Besides, a tool to extract a topological map from the grid map is presented. The main limitation of this approach is the high computational memory requirements, which prevents its use in large maps. The topological map will help to solve this memory issue of AgRobPP.

As future work, the most recent tools for feature-based localization (particle filter) needs to be validated in a real scenario. Also, the topological map will be integrated with AgRobPP in order to improve the computational efficiency of the path planning system in big dimension terrains. For the segmentation of satellite imagery, new data-sets will be tested with higher resolution images (drone, aeroplane or helicopter) to improve the path classification.

Acknowledgments. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020, and is co-financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 under the PORTUGAL 2020 Partnership Agreement, and through the Portuguese National Innovation Agency (ANI) as a part of project ROMOVI: POCI-01-0247-FEDER-017945.

References

1. Aguiar, A., Santos, F., Sousa, A.J., Santos, L.: FAST-FUSION: an improved accuracy omnidirectional visual odometry system with sensor fusion and GPU optimization for embedded low cost hardware. Appl. Sci. **9**(24), 5516 (2019)
2. Aguiar, A., Santos, F., Santos, L., Sousa, A.: A version of Libviso2 for central dioptric omnidirectional cameras with a laser-based scale calculation. In: Iberian Robotics Conference, pp. 127–138. Springer (2019)
3. Bailey, T., Durrant-Whyte, H.: Simultaneous localization and mapping (SLAM): part II. IEEE Robot. Autom. Mag. **13**(3), 108–117 (2006)
4. Auat Cheein, F., Steiner, G., Perez Paina, G., Carelli, R.: Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. Comput. Electron. Agric. **78**(2), 195–207 (2011)
5. Cheein, F.A.A., Carelli, R.: Agricultural robotics: unmanned robotic service units in agricultural tasks. IEEE Ind. Electron. Mag. **7**(3), 48–58 (2013)
6. Deans, M.C., Hebert, M.: Bearings-only localization and mapping. Ph.D. thesis, Citeseer (2005)
7. dos Santos, F.B.N., Sobreira, H.M.P., Campos, D.F.B., dos Santos, R.M.P.M., Moreira, A.P.G.M., Contente, O.M.S.: Towards a reliable monitoring robot for mountain vineyards. In: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, pp. 37–43. IEEE (2015)
8. Dos Santos, F.N., Sobreira, H., Campos, D., Morais, R., Moreira, A.P., Contente, O.: Towards a reliable robot for steep slope vineyards monitoring. J. Intell. Robot. Syst. **83**(3–4), 429–444 (2016)
9. Duarte, M., dos Santos, F.N., Sousa, A., Morais, R.: Agricultural wireless sensor mapping for robot localization. In: Robot 2015: Second Iberian Robotics Conference, pp. 359–370. Springer (2016)
10. Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., Scaramuzza, D.: Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. J. Field Robot. **33**(4), 431–450 (2016)

11. Fernandes, E., Costa, P., Lima, J., Veiga, G.: Towards an orientation enhanced astar algorithm for robotic navigation. In: 2015 IEEE International Conference on Industrial Technology (ICIT), pp. 3320–3325, March 2015
12. Goto, T., Kosaka, T., Noborio, H.: On the heuristics of a* or a algorithm in its and robot path-planning. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), vol. 2, pp. 1159–1166, October 2003
13. Hahnel, D., Burgard, W., Fox, D., Fishkin, K., Philipose, M.: Mapping and localization with RFID technology. In: 2004 Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004), vol. 1, pp. 1015–1020, April 2004
14. Karaman, S., Walter, M.R., Perez, A., Frazzoli, E., Teller, S.: Anytime motion planning using the RRT*. In: 2011 IEEE International Conference on Robotics and Automation, pp. 1478–1483. IEEE (2011)
15. Kurt-Yavuz, Z., Yavuz, S.: A comparison of EKF, UKF, FastSLAM2. 0, and UKF-based FastSLAM algorithms. In: 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), pp. 37–43. IEEE (2012)
16. Mendes, J., dos Santos, F.N., Ferraz, N., Couto, P., Morais, R.: Vine trunk detector for a reliable robot localization system. In: 2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 1–6, May 2016
17. Mendes, J.M., dos Santos, F.N., Ferraz, N.A., do Couto, P.M., dos Santos, R.M.: Localization based on natural features detector for steep slope vineyards. J. Intell. Robot. Syst. **93**(3), 433–446 (2019)
18. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: AAAI/IAAI, pp. 593–598 (2002)
19. Moreira, A.P., Costa, P., Costa, P.: Real-time path planning using a modified a* algorithm. In: Proceedings of ROBOTICA 2009-9th Conference on Mobile Robots and Competitions (2009)
20. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recogn. **29**(1), 51–59 (1996)
21. Pinto, R., Santos, F.N., Sousa, A.J.: Robot self-localization based on sensor fusion of GPS and iBeacons measurements. In: 11th Edition of Doctoral Symposium in Informatics Engineering (DSIE 2016) (2016)
22. Pivtoraiko, M., Knepper, R.A., Kelly, A.: Differentially constrained mobile robot motion planning in state lattices. J. Field Robot. **26**(3), 308–333 (2009)
23. Raja, P., Pugazhenthi, S.: Optimal path planning of mobile robots: a review. Int. J. Phys. Sci. **7**(9), 1314–1320 (2012)
24. Reis, R., Mendes, J., dos Santos, F.N., Morais, R., Ferraz, N., Santos, L., Sousa, A.: Redundant robot localization system based in wireless sensor network. In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 154–159, April 2018
25. Rodriguez, S., Tang, X., Lien, J.-M., Amato, N.M.: An obstacle-based rapidly-exploring random tree. In: Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), pp. 895–900. IEEE (2006)
26. Santos, A., Cunha, A., Macedo, N., Lourenço, C.: A framework for quality assessment of ROS repositories. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4491–4496, October 2016

27. Santos, L., Ferraz, N., dos Santos, F.N., Mendes, J., Morais, R., Costa, P., Reis,R.: Path planning aware of soil compaction for steep slope vineyards. In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 250–255, April 2018
28. Santos, L., Santos, F.N., Magalhães, S., Costa, P., Reis, R.: Path planning approach with the extraction of topological maps from occupancy grid maps in steep slope vineyards. In: 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 1–7, April 2019
29. Santos, L., dos Santos, F.N., Mendes, J., Ferraz, N., Lima, J., Morais, R., Costa, P.: Path planning for automatic recharging system for steep-slope vineyard robots. In: Iberian Robotics Conference, pp. 261–272. Springer (2017)
30. Santos, L., Santos, F., Mendes, J., Costa, P., Lima, J., Reis, R., Shinde, P.: Path planning aware of robot's center of mass for steep slope vineyards. In: Robotica, pp. 1–15
31. Santos, L., Santos, F.N., Filipe, V., Shinde, P.: Vineyard segmentation from satellite imagery using machine learning. In: EPIA Conference on Artificial Intelligence, pp. 109–120. Springer (2019)

From Control to Coordination: Mahalanobis Distance-Pattern (MDP) Approach

Shuichi Fukuda(✉)

Keio University, SDM, 4-1-1, Hiyoshi, Kohoku-Ku, Yokohama 223-8526, Japan
shufukuda@gmail.com

Abstract. It is pointed out in this paper that we have been discussing intelligence with focus on control, but coordination becomes increasingly important. In other words, intelligence up to now has been knowledge-based, but now we need to be wisdom-focused, because changes take place frequently and extensively and in an unpredictable manner. We need to interact directly with the outside world to adapt to such rapidly changing environments and situations. To describe it another way, we need to go ahead by trial and error. Thus, Pragmatic approach, PDSA, plays an important role. But to learn from failures, we need a quantitative performance indicator to improve. It is pointed out in this paper that non-Euclidean Distance, Mahalanobis Distance (MD), which does not require orthonormality and units enables simple expansion of dimensionality and easy and fast processing. So, it serves as an excellent quantitative performance indicator. Further, if we introduce pattern approach and integrate MD and pattern, then we can have a performance indicator, which is not only quantitative, but also holistic.

Keywords: Coordination · Balancing · Performance indicator · Non-Euclidean approach · Mahalanobis Distance · Pattern · Robot human teamworking · IoT

1 World is Changing

Knowledge is our structured experience. Knowledge is effective when the environments and situations do not change or change in a predictable manner. In short, when the changes are static, knowledge plays a very effective role.

But changes yesterday and today are completely different. Changes were smooth yesterday, so we could differentiate them and predict the future. We could understand how the environments and situations change. But today changes are sharp. So, we can no more differentiable them and predict the future (Fig. 1).

Thus, "Adaptability" becomes important. How we understand the current environment and situation becomes crucially important.

In addition to changing changes, our world is quickly expanding (Fig. 2). Yesterday, our world is small and closed with boundary, but now it becomes open and the boundary disappears. In a closed world with boundary, we can apply rational approaches. Thus, our knowledge worked very effectively.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 286–302, 2021.
https://doi.org/10.1007/978-3-030-55180-3_22

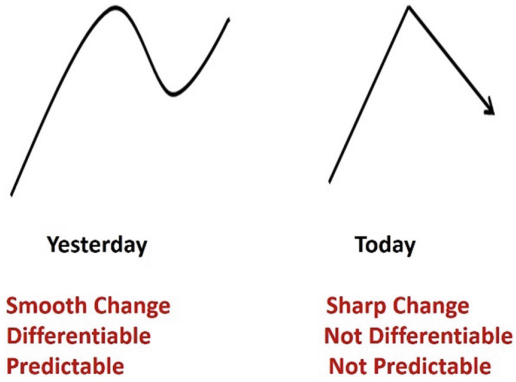


Fig. 1. Changes yesterday and today

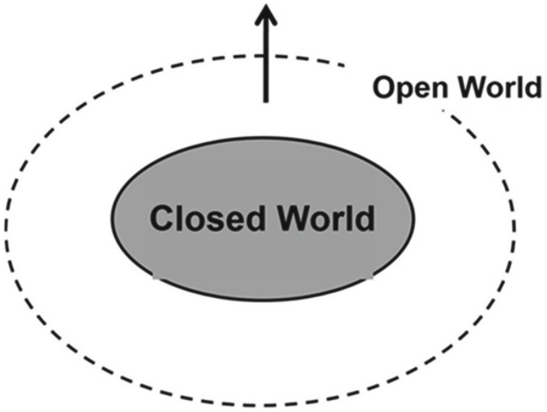


Fig. 2. Closed world to open world

But in an open world, we cannot apply rational approaches in a straightforward manner. We needed some idea to expand rational approaches to a quickly expanding world. This is the idea of "System Identification". Its basic idea is the same as how we identify the name of a river. If we look at the river flow, it changes every minute so we cannot identify its name. But if we look around, we can find mountains, forests, etc. that do not change. So, we can identify its name easily (Fig. 3).

Until very recently, although our world expanded very quickly, changes did not change very much. We still could foresee the future. Thus, by entering the known input, and observe the output (Fig. 4), we could expand the rational approaches and established the controllable world (Fig. 5).

But our world expanded more and more rapidly, and changes are becoming unpredictable. In other words, until very recently we could be on the bank. But now we are thrown into the water. As the flow changes every minute and there are no feature points to apply rational approaches, the only way for us is to make decisions by trial and error.

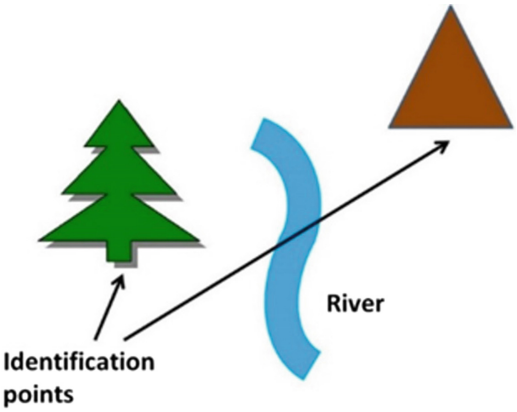


Fig. 3. How can we identify the Nname of a river?

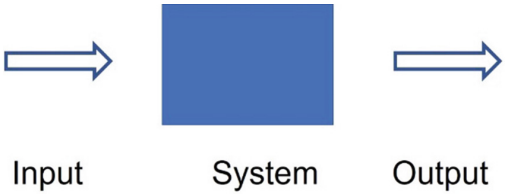


Fig. 4. System identification

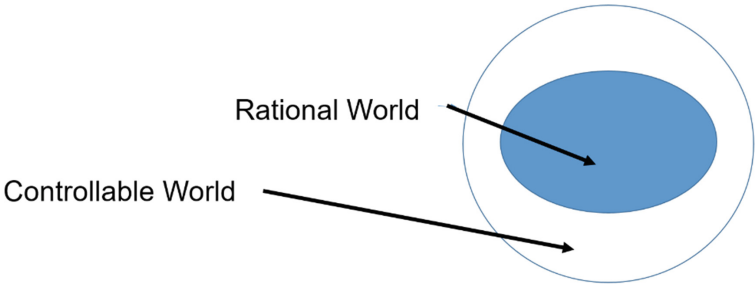


Fig. 5. Rational world to controllable world

To put it another way, in a predictable world, our focus has been on control. How we can get to the goal faster and more effectively, because the goal and the track are clearly defined from the beginning. But today, environments and situations change frequently and extensively and in an unpredictable manner, so there is no other way than trial and error approach. In other words, learning from failures becomes important. We need to interact directly with the outside world in order to understand how we should move forward.

2 Increasing Importance of Coordination

Thus, control has been very important in our engineering up to now. It may not be too much to say that our engineering has been rigid and static. Our world up to now has been a controllable world.

But when the environments and situations change frequently and extensively and in an unpredictable manner, we need to perceive correctly and put all pieces of information and tools together in an adaptive way to cope with the changes of the outside world.

Control is more associated with our brain, but coordination is more associated with our body, because it is our body that directly interacts with the outside world. In this sense, we must learn from invertebrate. The octopus represents invertebrate and it tells us how we should directly interact with the outside world. Octopus parents die immediately after their babies are born. So, there is no transfer of knowledge or past experience from generation to generation. They have no other way than to live only on their own instinct.

Figure 6 shows the Evolution Tree. Octopus and human are on the opposite side of the evolution tree. Although the octopus brain looks almost the same size as human's, its role is to coordinate its arms and body. Human brain, on the other side, is to process information.

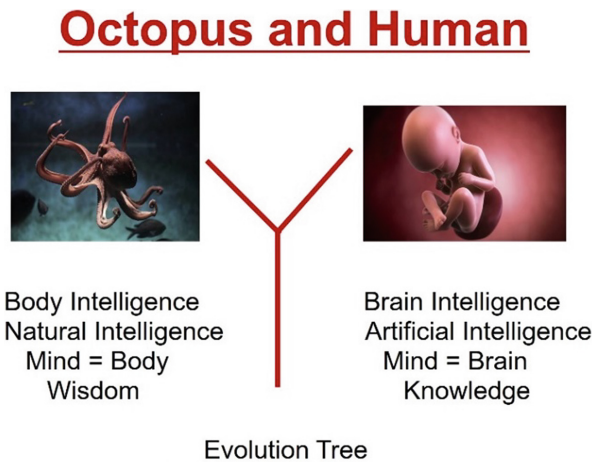


Fig. 6. Evolution tree – octopus and human

We may call our intelligence Brain Intelligence and octopus's Body Intelligence. Or we may call "Brain Intelligence" *Artificial Intelligence* and "Body Intelligence" *Natural Intelligence*.

Mind-Body-Brain issue is often discussed. But we should remember that when we make decisions, we say "We make up our mind". We do not say "We make up our brain". Then, what does mind mean?

Figure 7 shows my interpretation of "Mind". Indeed brain plays an important role in decision making as it constitutes mind together with body. But such example of reflex indicates not all information is transferred to brain. Some of them are processed at the level of body. And it is our body that directly interacts with the outside world.

But when we discuss decision making, we often forget about the role of body. We only focus our attention on brain. The octopus teaches us how important instinct is to directly interact with the outside world. In fact, the octopus is known as "Expert of Escape". Even if we screw them up in a container, they make many trials and errors and escape. If we are locked up in a screwed container, we would be panicked and may not be able to escape.

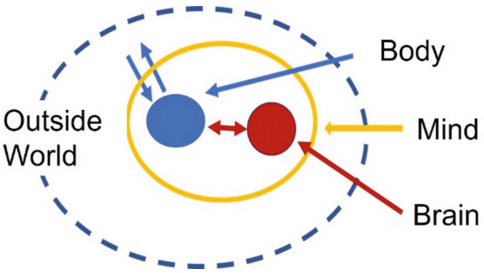


Fig. 7. Mind-Body-Brain

As we are now in the water and we need to make decisions by trial and error, we should learn from the octopus and we should utilize our instinct more. This may be called Wisdom. Our engineering will be changing from Knowledge-based to Wisdom-focused.

Figure 8 shows Nikolai Bernstein's cyclogram of hammering. Bernstein is well-known in human motion control and he pointed out that human motion trajectories vary widely from time to time at first, but it is fixed when we get close to the target object.

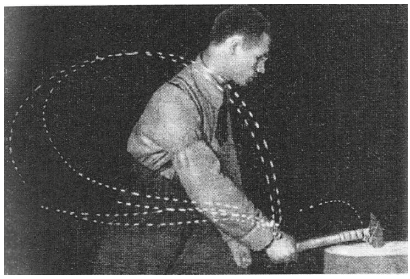


Fig. 8. Nikolai Bernstein's cyclogram of hammering

He pointed out the tremendously large degrees of freedom is the big issue in human motion control. He also pointed out the importance of coordination in human motion control in his book [1].

At the latter stage when our trajectory is fixed, it is nothing other than the issue of control. We have a clear goal. But at the first stage when our trajectories vary widely, we coordinate our body parts and balance our body. The body parts used vary from time to time. So, we need to find out what parts we should use and we coordinate by trial and error.

Further, trajectories vary from time to time not only with respect to one person in order to coordinate our body parts and balance our body, but also from person to person, because our body builds are different from person to person. Thus, there is no single rule for control.

Interestingly, most robots are controlled. We give instructions from outside and robots respond. And when we discuss humanoid robots, it is modeled based upon our skeleton model as shown in Fig. 9.

Fig. 9. Humanoid robot

But if we consider such a robot which works together with a human (Fig. 10), we should consider that our motions vary from person to person and even with a single person, from time to time, as Bernstein made clear. Thus, the degrees of freedom are tremendously large, and we need approaches other than rational ones for coordination.

Fig. 10. Robot working together with human

Interestingly enough. Our muscles harden when we finalize our trajectory [2]. Thus, if we note this latter portion of our motion, then skeleton model works well and we can apply rational approaches. That is the basic concept of humanoid robots.

But we need to remember babies swim in amniotic fluid before they are born. And when we coordinate our body's parts and balance our body, we use muscles rather than skeleton. We use these muscles to swim in the water. But after we are born and start walking on earth, our world changes and we start to move in gravitational field. And muscles closely related to skeleton works together with bones. But in the water or in amniotic fluid, we use other kinds of muscles for coordination and balancing [3, 4].

To achieve such movements, we need other approaches which can accommodate large degrees of freedom, but still can provide a performance indicator to improve our motion. This is nothing other than pragmatic approaches. In Pragmatism, learning from failures plays an important role. Fail here means not failures, but in the sense that our model fails our expectations. Thus, as Fig. 11 shows, which is known as PDSA, Plan (we think of a hypothesis) → Do (apply to the problem) → Study (observe if it works) → Act (if it works, then use it. Otherwise, repeat this process until we find out the working hypothesis).

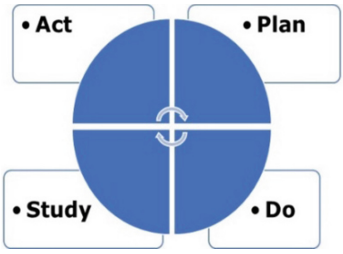


Fig. 11. PDSA (Plan-Do=Study-Act) cycle

To repeat this cycle and find the appropriate hypothesis, we need a performance indicator to judge the hypothesis or a trial satisfies our expectations.

To describe it from another perspective, this is nothing other than satisficing. Herbert Simon coined the word "satisficing", which is satisfy + suffice [5]. He introduced this idea because the problem of computational complexity comes up with increasing dimensionality. So, we cannot optimize the solution and we need to accept, if the result is satisfactory enough. Pragmatism is a way of thinking how we can find a satisficing results.

In fact, in simulated annealing in global optimization, we repeat the process many times and after many trials, we regard the highest peak as the highest (the optimum), but higher peak might follow immediately after. No one knows. We just make ourselves satisfied by repeating the cycle many times.

But to swim in the water, we need to know how better we are doing this time or how we can improve. If no performance indicator is provided, then, trials and errors end up as unorganized. To learn to swim, we need to organize our trials and errors as shown in Fig. 12. In order to organize appropriately, we need a holistic performance indicator.

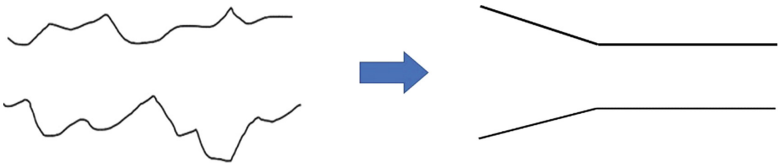


Fig. 12. Organized trials and errors.

3 Mahalanobis Distance (MD)

The Mahalanobis Distance (MD) a measure of the distance between a point P and a distribution D, introduced by P. C. Mahalanobis in 1936. It is a multi-dimensional generalization of the idea of measuring how many standard deviations away P is from the mean of D [6] (Fig. 13).

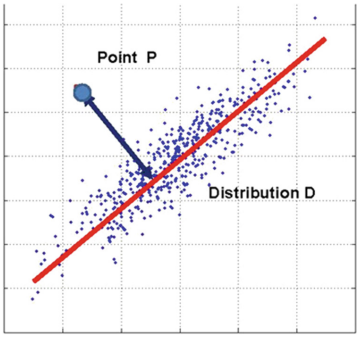


Fig. 13. Mahalanobis Distance (MD)

Although MD is originally proposed for multivariate statistics, let us think much simpler. Our purpose is not to discuss rigorous multivariate statistics, but to utilize it for learning how to control motions in much easier and in more self-convincing way.

MD is unique because it normalizes distance from the mean by standard deviation. So, it is unitless and can be extended to multi-dimension without considering orthonormality.

Mahalanobis proposed MD to find out the outliers, because the outliers cause serious problems in statistical analysis. He was interested in Principal Component Analysis (PCA). Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

The primary interest of Mahalanobis was to remove the outliers, because outlines affect the dataset, so that principal component may not be correctly obtained (Fig. 14).

Further Principal Components have orthogonal relations with each other (Fig. 15). As PCA is based on orthonormality, Euclidian Distance can be used. To rigorously discuss PCA, Euclidian Space and Euclidian Distance are useful. But units come with

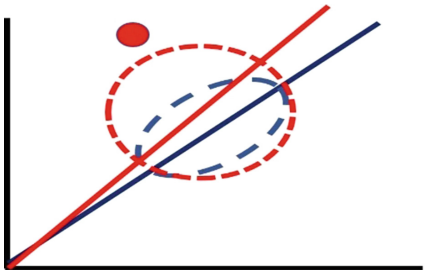


Fig. 14. Point affecting PCA

them. In fact, the purpose of multivariate analysis is to represent multivariate data in fewer dimensions. And eigenvectors describe directions in space.

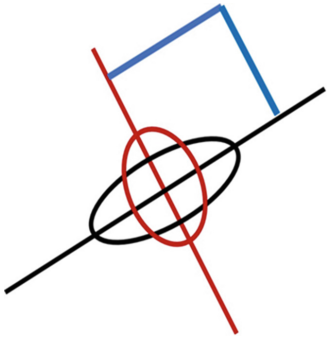


Fig. 15. Principal components are orthogonal to each other

But if the main purpose is to remove outliers, we do not need to stick to Euclidean Distance. We can introduce a much simpler approach. That is MD.

What matters is how much far away the point is from other observed data. If MD is too large, it means that it is far away from other observed data. So, possibly it is an outlier. Such outliers may be due to some errors or due to some other reasons and it is not appropriate to consider them as good and reliable data.

Just to remove the outliers from good dataset, it is enough to know standard deviation/mean. If this is too large, it means that the point is too far away from other data. So, it can be regarded as an outlier and can be removed. This is very simple and does not call for any orthogonality. Thus, it can be easily extended to multidimensional applications and further we do not have to care about the unit. So, multidimensional unitless metric approach can be established.

Although the original aim of Mahalanobis is considered to be just to simplify the procedures of finding out outliers, MD is coming to be used more and more today, because in addition to its simplicity, due to its unitless feature, it can be used for many different cases where each sample data is composed of many different units [12].

4 Pattern

4.1 Mahalanobis Taguchi System (MTS)

Taguchi realized that although Mahalanobis was interested in finding outliers and to exclude them to secure good dataset, he can utilize MD in a different way if he introduces patterns. Then Taguchi's definition of quality can be evaluated very easily.

Taguchi's quality definition is unique. His quality is based on Loss. Usually when we discuss quality, we are thinking how better functions can be added. But Taguchi's definition is substantially the same as cost/performance and how we can reduce cost. He introduced the idea of Loss Function [7] to reduce the cost of flow in production.

Another point characterizing his Loss Function is he considers the Loss of the Society. Industries emphasize the importance of cost reduction. But if the reduction is made at the expense of social loss, then the company cannot continue their business in the long run. So, he emphasized the importance of considering loss from the society perspective. In other words, his Loss Function is how to minimize the energy consumption and it truly matches the concept of sustainability.

Taguchi is very well known for his Taguchi Method. Original Taguchi Method uses orthogonal table and it is orthonormality-based. But he introduced his original S/N ratio idea. This is very much different from S/N ratio used elsewhere. To describe Taguchi's S/N Ratio in a simple way, it is the distance of P from the Distribution of Noise. Roughly speaking, his basic idea of Variance/Mean is the same as MD. But Mahalanobis used it for removing outliers, but Taguchi used MD for securing quality.

If this figure is large, we need to adjust parameters to stay within the acceptance level. If the figure is within the acceptance level, then we do not need to pay any special attention to each parameter.

To describe this in another way, if the parameters are represented as patterns, and the pattern is within the acceptance level, then quality is good. Taguchi realized that if he uses MD, then he does not have to care about orthonormality and as MD is unitless, he can introduce pattern approach to evaluate quality.

Figure 16 illustrates the idea. Unit space is created based on typical samples (Fig. 17) and MD is calculated for each quality pattern. If a MD of the pattern exceeds the threshold value, then that pattern is not acceptable.

We should note that we use pixels to produce a pattern (Fig. 18), so that any images can be represented as patterns.

This idea was very much welcomed by industries who cannot control quality element by element. What is better, this approach does not call for orthogonal arrays. So, it is quite easy for them to control quality, because it is not on an element basis but on the whole product or production basis. Thus, Malahanobis Taguchi System [8] spread very widely in a very short time. So, Mahalanobis Taguchi System is now being applied in many different fields.

It should be pointed out that Euclidean space based Taguchi Method is based on the idea of "Control", but MTS is how we can coordinate different quality-related elements and balance them to secure quality. So, it is "Coordination".

We must remember that Taguchi's definition of quality is totally different from the usual definition of quality. When we say, quality, we think of how we can have better

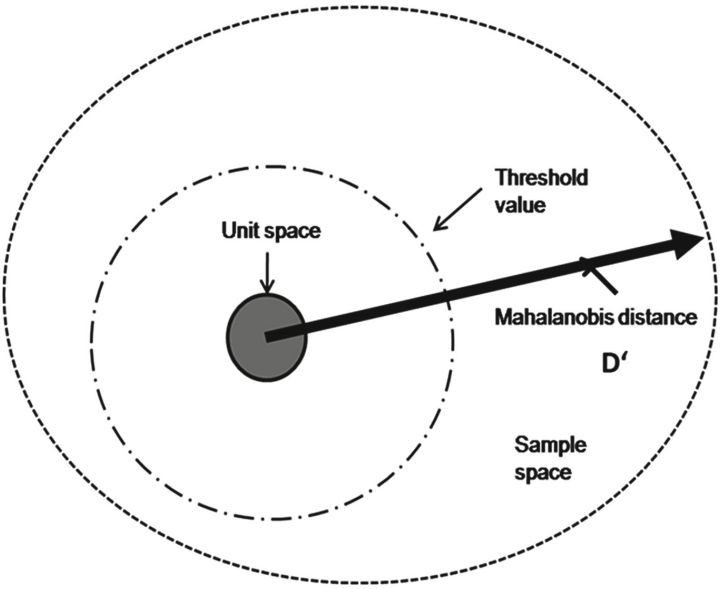


Fig. 16. MD based pattern identification

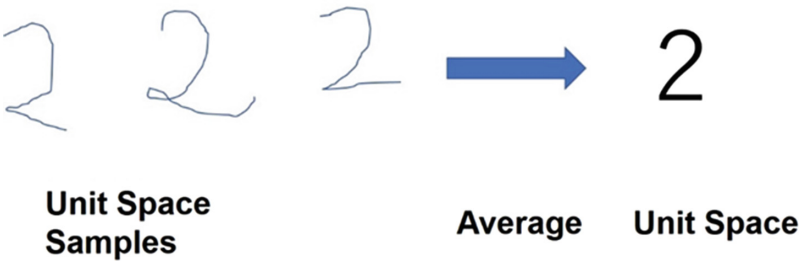


Fig. 17. Defining unit space

working products. But Taguchi's definition is how we can reduce the loss in production. So, he introduced Loss Function to evaluate quality. When the Loss Function is reduced to the minimum, then quality is secured (Fig. 19). If we compare, Fig. 12 and Fig. 20, we will understand why Taguchi noticed MD.

This may sound very strange to quality control engineers, but this is very popular in industry. Industry emphasizes cost/performance. Taguchi's assertion is, in other words, to reduce the cost as much as possible. What differentiates his assertion from industries is he is not thinking industry alone. He is thinking about Society. Even If one industry succeeds in achieving the best cost/performance for that particular company, but if it introduces large cost to the society, then that strategy would not last long and from the social point of view, it is not controlling quality in the appropriate way. The loss should be minimized from the social point of view. This is his assertion.

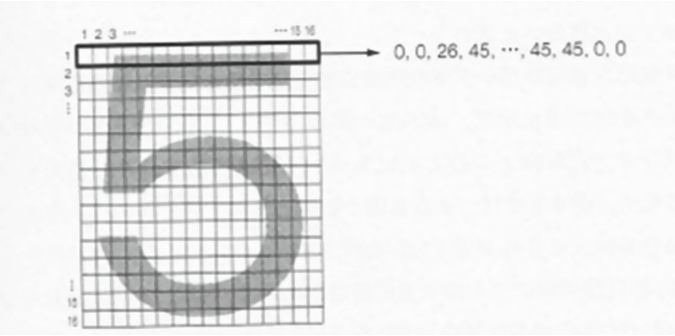


Fig. 18. Representing images as a pattern

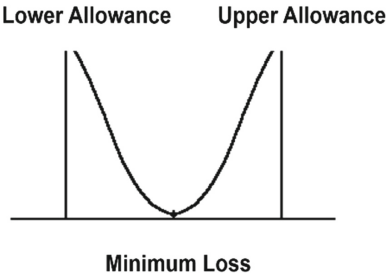


Fig. 19. Loss function

Thus, pattern approach using MD is widely introduced in many fields, not only in quality control, but in other fields as well.

Let us consider Taguchi's Approach of Loss Function from another perspective. It is not too much to say that Taguchi knows human psychology very well.

Of course, everybody would like to have better products, but once he or she has it, he or she would like to have one more step better product. So, they continue wanting better products. But of course, that requires a lot of money and efforts, but such investments are not always successful. If the customers are not satisfied, then what a large damage that industry will suffer. But even if your product remains the same in terms of functions, etc. you will feel satisfied, if the quality remains the same, no matter under what situations you are using them.

It is not a homerun, but a small hit. But if the player strikes a hit every time, even though it may be small, then you can trust that player. This is the problem of trust and it is nothing other than establishing brand for the company.

As the idea of Loss Function includes Loss at the Society level, we cannot introduce orthogonal arrays easily. We cannot carry out Design of Experiments with the Society included. The problem is too much complicated and complex. Thus, the idea of the Loss Functions serves to secure trust and it leads to establishing brand, Taguchi's pattern-based MD approach is, therefore, widely accepted by a large number of industries.

4.2 Detection of Emotion from Face

Fukuda noticed Taguchi's pattern approach because he used to study detection of emotion from face. His group tried many different image processing techniques, but they were too much complicated and took a lot of time. But the results were not satisfactory. Then, he realized that we can understand emotions of the characters in cartoons without any difficulty.

At that time, cartoons were black and white, and characters were very simple. But we can understand their emotions at once. So, he introduced the cartoon face approach and his group succeeded in detecting emotion from face easily in a very short time (Fig. 20) [9, 10].

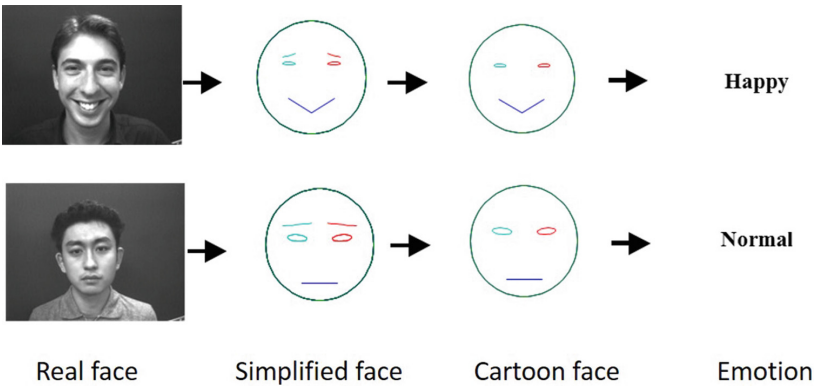


Fig. 20. Cartoon face model

At that time, Fukuda team used Cluster Analysis to group datasets. And in this series of research, they found out that acceleration plays an important role for emotional detection. Thus, they realized the importance of dynamic approach.

When Fukuda came across Taguchi's pattern approach using MD, he realized that as MD and Cluster analysis share the same idea, he can introduce Taguchi's pattern approach to motion control. But Taguchi was interested only in static pattern identification. But the problem here is dynamic. So, Fukuda extended Taguchi's approach to dynamic pattern recognition or it is better to say to dynamic pattern creation.

Vlaho Kostov wrote another paper, which utilizes an electromagnetic tool to extract motion. This is not an emotion-related study, but it will illustrate how such tools can be used for coordinating motion [11].

5 MD Pattern Approach (MDP) to Motion Coordination

Mahalanobis Distance (MD)-Pattern Approach (MDP) Fukuda developed is the extension of MTS. Taguchi develop MD-Pattern approach, but it is for pattern matching, so it is static.

Fukuda extended it to dynamic issue of motion coordination. Interestingly enough, although the same issue of human movement control is taken up, it is called motion control when we observe from outside. And it is called motor control, if we pay attention to inside of our body. As Bernstein's cyclogram indicates, the latter movement trajectory is fixed. So, it is straightforward to apply rational approaches. It boils down to the control problem. But in the first portion, our trajectories vary very widely. Therefore, research on "Coordination" is important.

If we only observe trajectories from outside, our movements can be captured as images and we can evaluate our performance quantitatively using MD and patterning images (Fig. 19). We must note we need to process not only the image of the body position, but also the image of the body changes (speed, acceleration). We can use MDP approach in this way, but only the change of body positions can contribute a great deal toward improving our trials. We can leave the decision to humans how he should improve his performance based on this information. Anyway, we need to compare the images of the trial this time and previous ones and understand how we should improve. Thus, in MTS, MD gives the criterion whether the pattern matches or not, but in Fukuda's MDP, the image patterns are compared to study whether the difference between this time and the previous times decreases or not. If it does not decrease, then we need to try another movement. If it does, then we can keep on improving that way.

Regrettably enough, although research on motor control is being carried out in many different areas, but how inside of our body works for motor control as a whole is not clear yet. So, at the present time, we have no other choice but to use motion control data and leave the decision how we should improve our movement to instinct.

But as invertebrate, such as the octopus, demonstrates, we need to use our instinct more to directly interact with the outside world. So, in this sense, the way MDP utilizes our instinct may be reasonably acceptable.

6 Applications

MD Pattern Approach (MDP) works very well for such application as shown in Fig. 21. As mentioned before many times, MDP can solve the problems whose parameters cannot be identified. Swimming is the typical example. But if we remember that sensing fibers are being developed, it is easy to develop a shirt which expands when it is hot and which shrinks when it is cold. It may be called a very personal air conditioner. In artificial legs or arms, the interaction between these artifacts and human bodies is very often discussed. But as our body builds are different from person to person and how we feel is different also from person to person, such a tool as MDP will serve a great deal to find the best break-in conditions so that patients can move in their own way. This would bring them great happiness. Such careful attention towards individual characteristics would be taken care of by MDP very well so that it will bring satisfaction and happiness to patients in rehabilitation and in healthcare.

It should also be emphasized that MDP will contribute to create physical arts. Figure skating is a typical example. But all sports are in essence physical art. So MDP will serve a great deal for creating physical arts and for enjoying them (Fig. 22).

Fig. 21. Wearable robots

Fig. 22. Physical arts

7 11 Best to Best 11

Kute Rockne emphasized the importance of "11 Best to Best 11" He points out that even if we form a team with 11 best players, we cannot make up the best team. The best team is composed of 11 players who understand the changes of the game flow and can play adaptively.

To illustrate his words in another way, the goal was clear yesterday so that our world was tree-structured. Each member is expected to work best in his role. But with the rapid shift to frequent and extensive unpredictable changes, we need to adapt to such changes. Therefore, our world shifted to network-structure, because in a network, all nodes can be an output node. But until recently, fixed network worked well, but the changes of environments and situations become more and more frequent and extensive, so that we need truly "Adaptive Network". In short, "Coordination" becomes crucially important (Fig. 23).

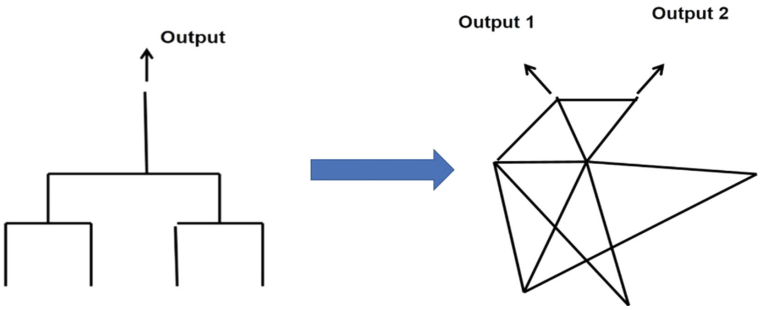


Fig. 23. Tree to network

As Fig. 24 shows the change of our position from yesterday to today. Yesterday, we instructed machines from outside and they responded. But today, we work together with machines in the system.

If we take up soccer, we can understand this shift. Yesterday, each player was expected to do his best at his position and formation did not change during the game. Thus, the manager can be off the pitch and give instructions from outside.

But today, games change very frequently and extensively so that a team need to play in a different formation to adapt to the rapid changes and to win. Therefore, the manager needs to play on the pitch to understand the rapidly changing game flow in order to organize a team in a very flexible and adaptive way. Thus, midfielders are now playing-managers in soccer.

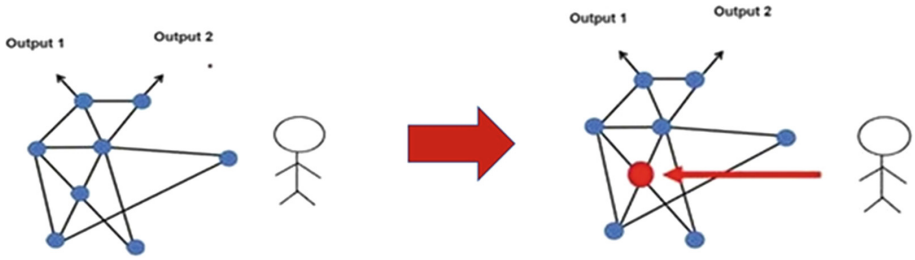


Fig. 24. Outside to inside

Figure 25 illustrates why Coordination becomes more important than Control. Up to now, changes have been predictable and we can identify system parameters. Thus, our world has been explicit and verbal. Therefore, we could apply rational approaches.

But today, changes become unpredictable and our world becomes tact and nonverbal. Therefore, how we adapt to such frequent and extensive unpredictable changes becomes crucially important. To achieve such a challenge, we need more heads or more things. In short, nothing can be achieved without teamworking. In order to maximize the effect in teamworking, team members need to communicate in a proactive manner or to be ready in advance for the next network formation.

Control
Predictable Changes
System Identification Parameters
Explicit, Verbal → Rational Approach

Coordination
Unpredictable Changes
Tacit, Nonverbal
Adaptability
Teamworking
Communication *Proactive*

Fig. 25. Control to coordination

Figure 26 shows traditional to next generation engineering.

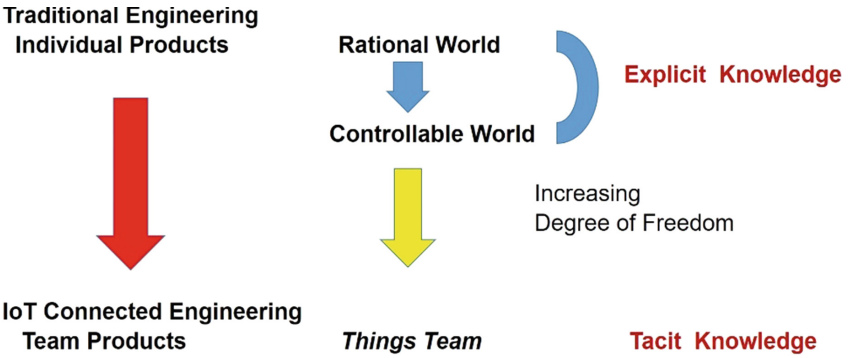


Fig. 26. Traditional to next generation engineering

8 Summary

"Coordination" becomes increasing important to adapt to the frequent and extensive unpredictable changes today. To organize an appropriate adaptive network, we need a holistic and quantitative performance indicator, because we have no other choice but to go ahead by trial and errors and we need some performance indicator to improve our trials.

Mahalanobis Distance (MD) - Pattern Approach is proposed for coordination. Its benefits are free from orthonormality and unit requirements so that any number of dimensions can be processed easily and in a very short time.

References

1. Bernstein, N.: Co-ordination and Regulation of Movement. Pergamon Press, Oxford (1967)
2. https://www.nature.com/articles/srep45486
3. Kumamoto, M.: Bi-articular muscle unique control properties supported by biological evolutionary evidence. J. Robot. Soc. Jpn. **28**(6), 660–665 (2010). (in Japanese)
4. Kumamoto, M.: Motor control properties induced by bi-articular muscles. Jpn. J. Rehabil. Med. **19**, 631–639 (2012). (in Japanese)
5. https://en.wikipedia.org/wiki/Satisficing
6. https://en.wikipedia.org/wiki/Mahalanobis_distance
7. Ross, P.J.: Taguchi Techniques for Quality Engineering: Loss Function, Orthogonal Experiments, Parameter and Tolerance Design. McGraw-Hill Professional, New York (1995)
8. Taguchi, G., Chowdhury, S., Wu, Y.: The Manalanobis Taguchi System. McGraw-Hill Professional, New York (2000)
9. Kostov, V., Yanagisawa, H., Johansson, M., Fukuda, S.: Method for face-emotion retrieval using a cartoon emotional expression approach. JSME Int. J. **44**(2), 515–526 (2001)
10. Kostov, V., Fukuda, S., Johansson, M.: Method for simple extraction of paralinguistic feature in human face. Image Vis. Comput. J. Inst. Image Electron. Eng. Jpn. **30**(2), 111–125 (2001)
11. Kostov, V., Fukuda, S.: Emotional coding of motion using electromagnetic 3D tracking instrument. Appl. Electromagnet. Mater. Sci. Device Jpn. J. Appl. Electromagnet. Mech. **8**, 229–235 (2001)
12. Fukuda, S.: Self Engineering: Learning From Failures. SpringerBriefs in Applied Sciences and Technology. Springer, Cham (2019)

Dynamic Proxemia Modeling Formal Framework for Social Navigation and Interaction

Abir Bellarbi[1,2,3(✉)], Abdel-illah Mouaddib[3], Noureddine Ouadah[2], and Nouara Achour[1]

[1] Université des Sciences et de la Technologie Houari Boumediene (USTHB), Bab Ezzouar, Algeria
a_bellarbi@cdta.dz

[2] Centre de Développement des Technologies Avancées (CDTA), Baba Hassen, Algeria

[3] Groupe de recherche en informatique, image, automatique et instrumentation de Caen (GREYC), Caen, France

Abstract. In this work, we address the problem of mobile robot social navigation in crowded environment. As a formal framework, we propose a new approach based on the proxemia principal. Since all previous works considered only static case, independently from the activity nature, the modality of interaction, the support of communication (phone, ...) and the spatial organisation of the group persons with which we interact. In this paper, we propose a formal framework of the "Dynamic Proxemia modeling approach" (DPMA) to produce a heat map of social navigation and interaction for a group of people. We validate it by experiments using ROS on simulation and on a Pionner robot in a mediation event example, which shows promising results.

Keywords: Social navigation · Human-robot interaction · Proxemia

1 Introduction

Actually, the robots are very present in our life to make it easier by providing services. Like the guide robot in an airport, the cleaning robot, the server robot and the assisting robot for elderly persons. These robots should navigate in crowded environments and interact with persons. So, it must be accepted by the people adapting a social behavior in order to respect the persons' comfort and safety rules.

In the interaction the robot approaches persons, but in navigation it avoids them. It has two different behaviors respecting the proxima rules. The proxemia, as defined by Hall et al., represents the study of the distance which is established between the persons during an interaction, it changes according to the culture of the country [1]. This distance is static. In an interaction using a speaker, the person does not need moving to interact, contrary to when she uses direct speech, the interaction distance is not the same. As well as, welcoming visitors with a sign of hand or serving coffee, the proxemia is depending on the type of

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 303–321, 2021.
https://doi.org/10.1007/978-3-030-55180-3_23

interaction. Some activities have targets, such as explaining posters, the poster must be targeted during interaction, which influences the proxemia, contrary to activities without targets. The Proxemia is depending on all these factors, it can not be static, it is dynamic and varies according to the nature of the activity.

In another hand, for interacting with a group, Kendon et al. defined three types of spatial organisations of the group: L-shape, vis-a-vis and side by side, called the F-Formation [2]. So, the type of F-Formation must be considered in the proxemia.

The previous works considered the proxemia static in interaction. Several studies were developed for social navigation. Pandey et al. used the human intention to adapt the robot behavior to the environment. Their work is based on the reflection and the robot behavior during the interaction [3]. Dewantara et al. used social forces to propose a guiding behavior [4], and Ramirez et al. used the Inverse Reinforcement Learning to teach the robot how and where to approach the person [5]. Lindner et al. defined modeled different spaces types, without formulating them [6]. Then, Papadakis et al. modeled the spatial interactions of person depending on context, and encoded it as a social map [7]. Svenstrup et al. modified the personal space by introducing the intention estimation of the person [8]. The latest works did not treat each person of a group separately, but sought associations among them to provide a social map [9]. Then, Vazquez et al. proposed a technique to control the orientation of the robot while interacting with a group of persons [10]. Chen et al. modified the algorithm of A* in order to avoid persons by using social rules as well as the position, orientation and displacement of persons [11]. Charalampous et al. considered the group activity (talking, walking, working) in the proxima, which is used in social navigation only [12]. Truong et al. developed an approach of Dynamic Social Zone, including the personal space, the type of interaction and the space of activity [13].

The cited works consider the navigation or the interaction separately and do not take into account the activity. Only the direct speech is considered in interaction. The personal space, the interaction space, the activity nature and the F-Formation were not formalized, nor taken into account together in navigation and interaction. Then, the proxemia is considered static for all activities. Bellarbi et al. proposed a new dynamic proxemia modeling approach DPMA, by considering all these factors [14].

Our contribution is to improve and complete the DPMA approach, by proposing its formal framework and validating it with experiments on the Pioneer robotic platform.

In the following, we define an example illustrating the scenario used in this work. Then, we present our contribution by formalizing the concepts in the navigation, and how they impact the space of interaction. We implement the approach to validate it on a real robot using ROS, and present the experiments results.

2 The Scenario

We choose a scenario of a scientific mediation event, the environment is represented by the Fig. 1. The robot should guide and assist the guests. First, it

announces the conference by using a speaker and welcomes the visitors at the entrance of the conference room. Then, it explains posters and serves coffee. It is represented by a gray circle, the tables by brown squares, the posters by yellow rectangles, the people in blue and pink. We assume that the persons are not moving. The perception gives the locations of the persons and the map of environment. The activity nature and the F-Formation type of a group are deducted from the locations of persons and the map of environment.

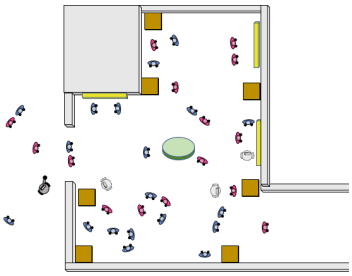


Fig. 1. Scenario of the activities

3 Formal Framework

3.1 Dynamic Proxemia

The dynamic proxemia is the study of the distance that is established between persons during an interaction, and influenced by the interaction modality, the interaction support, the activity nature and the group spatial organisation. The DPMA global diagram is described by the Fig. 2. The guide robot navigates in the environment and interacts with persons. There are two constraints: (i) the navigation constraints impacting the path planning, and (ii) the interaction constraints impacting the interaction location. In order to describe these constraints, we use cost functions. The selected path and interaction location are the ones with a minimal cost.

3.2 The Activity Model

The activity model is defined by a tuple s. t.:

$$Act(Modl, Supp, SAct, Move, Objc, Targ, Prio) \tag{1}$$

Where, "Act" is the activity, "$Modl$" is its modality, "$Supp$" is the interaction support in the case of an indirect activity, "$SAct$" is the activity space, "$Move$" if the activity needs a move action or not, "$Objc$" is the activity interaction object, "$Targ$" is the activity interaction target and "$Prio$" is the activity priority. The location of each activity is known on the environment map. Therefore, the activity nature can be deducted from the location of the person to interact with.

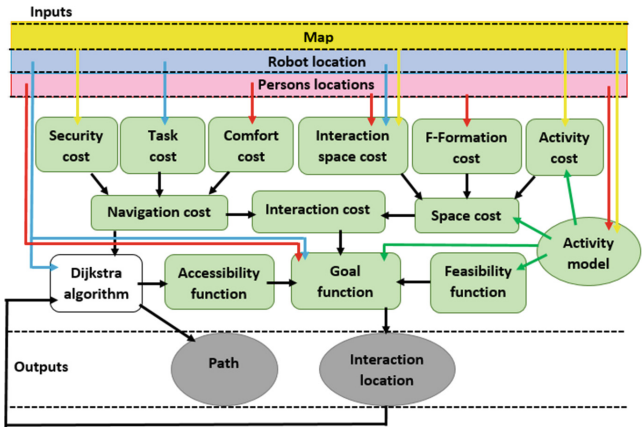


Fig. 2. The DPMA global diagram

3.3 The Feasibility of the Activity "*Feas*"

The activity is performed under some conditions. We study the feasibility of the activity $Feas$ defined by a logic proposition:

$$Feas(Act, Pos_R, Pos_P) = (Move \vee \overline{Move} \wedge (\overline{Supp} \wedge \overline{F_{Spa}(Pos_R)} \vee Supp \wedge \overline{F_{Act}(Pos_P)})) \quad (2)$$

F_{Spa} is the cost function of the space, F_{Act} is the activity space cost function, Pos_R and Pos_P are the positions of the robot and the person respectively.

The feasibility is verified if the activity requires a move $Move$, because it is up to the robot to find the goal position. In the opposite case, the robot should perform an activity without moving. If it is a direct interaction, we check if the cost function of the space for the robot position has a low value (ie it is an authorized position), or if it is an indirect interaction, we check if the cost function of the activity space for the person's position has a low value.

3.4 The Accessibility "*Accb*"

The inaccessible position, surrounded by obstacles, can not be Before a goal position G (represented by the vertice S_J). we must check all costs of the occupancy grid map cells M_{ij} (represented by the edge weight a_{IJ}), by using the navigation function F_{NavIJ}. Then, We compute the cost of each path W_z connecting the position Pos_R (represented by the vertice S_I) to S_J, using a classical path planning algorithm such as Dijkstra [15], $Dijkstra(S_I, S_J, F_{NavIJ})$.

The accessibility cost function F_{AccbIJ} of the cell S_J, checks the minimum navigation cost of the paths leading to this cell through the cells S_k. This cost is the minimum sum of the navigation costs of S_k cells. N_z is the number of paths for each cell and N_k is the cells number of the path for each cell S_J.

A G_{ij} goal position is considered accessible, if its accessibility condition $Accb_{G_{ij}}$ is checked (if there is at least one path leading to that cell with a low cost), described by Eq. 6. So, if the minimum navigation cost is below a threshold, we have:

$$Dijkstra(S_I, S_J, F_{NavIJ}) = \min_{z=1}^{N_z}\{W_1, ..., W_z, ...W_{N_z}\} \tag{3}$$

$$W_z = \{S_I, ..., S_k, ...S_J\} \tag{4}$$

$$F_{AccbIJ} = \min_{z=1}^{N_z} \sum_{k=0}^{N_k} F_{NavIJzk} \tag{5}$$

$$Accb_{G_{ij}} = \begin{cases} 1 \text{ if } F_{AccbIJ} < threshold \\ 0 \text{ else.} \end{cases} \tag{6}$$

3.5 The Navigation Cost Function F_{NavIJ}

The F_{NavIJ} is an $N \times N$ matrix, it contains real positive values if $I \neq J$, null values if $I = J$ or ∞ if there is no connection between S_I and S_J. It is computed from the function F_{NIJ}, which is a matrix $n \times m$ (n, m are dimensions of the occupancy grid map), includes the function of security cost F_{SecIJ} for obstacle avoidance, the function of task cost F_{TaskIJ} to choose the closest goal, and the function of comfort cost F_{ComfIJ}, to respect the social and security rules. These functions are $n \times m$ matrix.

$$F_{NIJ} = \alpha F_{SecIJ} + \beta F_{TaskIJ} + \gamma F_{ComfIJ} \tag{7}$$

where, α, β and γ are the positive weighting constants that sum up to 1, to prefer one criterion over the others.

Security Cost. For security, the selected path is the one with a minimum of clearance Clr_{ij}. This clearance is the clearance of obstacle for each cell M_{ij}, depending on the distance between the cell and the obstacles. The real positive values Sec_{ij} of the security cost function matrix F_{Secij} depends on the clearance Clr_{ij} or ∞ if this cell represents an obstacle Obs.

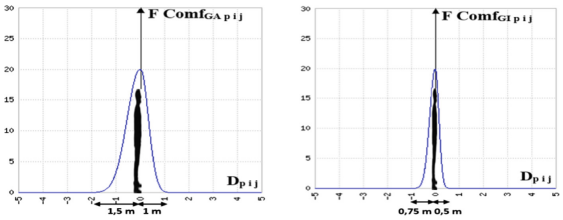
$$Sec_{ij} = \begin{cases} \infty & \text{if } M_{ij} = Obs \\ 1/Clr_{ij} & \text{else.} \end{cases} \tag{8}$$

Task Cost. For rapidity, the robot chooses the shortest path (commonly used in classical approaches) to reach the goal. So for each map's cell M_{ij}, the real positive values $Task_{ij}$ of the task cost function matrix F_{Taskij} is proportional to the Euclidean distance from M_{ij} to G_{ij}: $Dis(M_{ij}, G_{ij})$.

$$Task_{ij} = Dis(M_{ij}, G_{ij}) \tag{9}$$

Comfort Cost. For respecting the persons' comfort, the social constraints are considered in the choice of the path, to guarantee the persons' comfort. The robot should pass in front of persons and not behind them. So, for each person p, we have a comfort cost function, $F_{Comf_{pij}}$. The total comfort cost function F_{Comfij} is the sum of $F_{Comf_{pij}}$. We distinguished between two group types: GA is the group to avoid and GI is the group to interact with, and defined two cost functions for each type because the personal space of each person depends on the group type. So, this cost function is defined as follows:

$$F_{Comf_{pij}} = \begin{cases} F_{Comf_{GA_{pij}}} & \text{if } p \in GA \\ F_{Comf_{GI_{pij}}} & \text{if } p \in GI \end{cases} \tag{10}$$



(a) (b)

Fig. 3. Graphs representating the comfort cost for GA and GI

The cost functions $F_{Comf_{GA_{pij}}}$ and $F_{Comf_{GI_{pij}}}$ allow us to respect the personal space of people to avoid and to interact with respectively, as described in Fig. 3. So, for each map's cell M_{ij}, the real positive values $Comf_{pij}$ of these cost functions matrix are proportional to two half Gaussians which depend on the distance to the person D_{pij}. The Gaussian function is described by the Eq. 12, such as the X variable represents the Euclidean distance D_{pij} between M_{ij} and the person position Pos_p. The $Comf_{pij}$ values can be positive or negative depending on the orientation of the person θ_p. We determined the parameters of each cost functions, according to the personal space dimensions, as mentioned in Fig. 3.

$$D_{pij} = \begin{cases} Dis(M_{ij}, Pos_p) & \text{if } \theta_p = FRONT \\ -Dis(M_{ij}, Pos_p) & \text{if } \theta_p = REAR \end{cases} \tag{11}$$

$$Gaussian(X) = \frac{K}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X}{\sigma}\right)^2} \tag{12}$$

The total cost function F_{NavIJ} represents the cells total weight.

3.6 The Goal Point G

When the feasibility of an activity is verified, the goal cell is determined by Eq. (13):

$$Goal(Feas, Act, Accb_{M_{ij}}, Pos_R, Pos_P) = Feas \wedge (((Move \wedge Accb_{M_{ij}}) \times \min_{i=j=0}^{n,m} F_{Intij}) \vee \overline{Move} \times Pos_R) \quad (13)$$

Where, "$Goal$" is the goal point of the activity, "F_{Intij}" is the interaction cost function of the activity for each cell M_{ij}. The Goal cell of an activity $Goal(Feas, Act, Accb_{M_{ij}}, Pos_R, Pos_P)$ is computed if the feasibility of the activity conditions $Feas$ hold, and if the activity requires a move action ($Move$). The Goal cell is the cell accessible with the minimum cost of the interaction cost function $\min_{i=j=0}^{n,m} F_{Intij}$. In the opposite case ($\overline{Move}$), the Goal point is equal to the current robot position Pos_R.

3.7 The Interaction Cost Function F_{Intij}

For respecting navigation rules (security, comfort ...), the F_{Intij} includes a navigation cost F_{Nij} and a space cost function F_{Spaij}. These functions are represented by matrix $n \times m$.

$$F_{Intij} = F_{Nij} + F_{Spaij} \quad (14)$$

– The space cost F_{Spaij}:
When interacting with a group of persons ($Grp = True$), the F_{Spaij} is the F_{Spij} sum, for each person to interact with.

$$F_{Spij} = \alpha' \times (\overline{Supp} \vee Targ) \times F_{EIpij} + \beta' \times (Grp \wedge Move \wedge \overline{Supp}) \times F_{Formpij} + \gamma' \times F_{Actpij} \quad (15)$$

Where, "F_{EIpij}" is the cost function of the interaction space, "$F_{Formpij}$" is the cost function of the F-Formation, "F_{Actpij}" is the cost function of the activity and α', β', γ' are the positive weighting constants that sum up to 1, to prefer a criterion over the others.

The space cost function of each person F_{Spij} is the sum of two types of space functions under some conditions. However, when the activity is a direct activity ($\overline{Supp}$) or has an interaction target ($Targ$), the cost function of the interaction space F_{EIpij} is considered. When the activity is direct ($\overline{Supp}$), requires a move ($Move$) and the robot must interact with a group (Grp), the F-Formation cost function $F_{Formpij}$ is considered, while the activity cost function F_{Actpij} is considered in all cases.

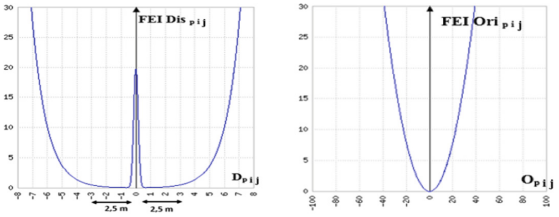
We will study the space cost function for each activity of the example. The functions of the interaction spaces F_{EIpij} and F-Formation $F_{Formpij}$ do not vary according to the activity, so we will formulate them in a general way to use them for all the activities. The activity space function F_{Actpij} varies according to the activity, so it is studied for each activity.

Interaction Space Cost: In order to interact with a person, the robot must reach the interaction space. This interaction space takes into account three criteria: distance, orientation and obstacles.

$$F_{EIpij} = F_{EIDispij} + F_{EIOripij} + F_{EIObspij} \tag{16}$$

Where, "$F_{EIDispij}$" is the interaction space cost function of the distance, "$F_{EIOripij}$" is the interaction space cost function of the orientation and "$F_{EIObspij}$" is the interaction space cost function of obstacles.

- *Interaction space cost "Distance":* The robot must not be too close or too far from the person, so we use the function $F_{EIDispij}$. Its values are real values of a Gaussian (Eq. 12) added to an exponential (Eq. 17), which depends on the absolute value of the distance D_{pij} ($X = |D_{pij}|$), between the person and the cell M_{ij} (Fig. 4a). This distance is defined above.



(a) (b)

Fig. 4. Graphs representating the interaction space cost "Distance" and "Orientation"

$$Exponential(X) = Ke^{(X-x_0)} \tag{17}$$

- *Interaction space cost "Orientation":* The robot must be in the vision field of the person, so we use the function $F_{EIOripij}$ depending on the orientation difference O_{pij} between the person and the cell M_{ij}. The values of $F_{EIOripij}$ are proportional to the O_{pij} orientation (Fig. 4b).

$$F_{EIOripij} = \begin{cases} \infty & \text{if } O_{pij} > \pi/2 \\ KO_{pij}^2 & \text{else.} \end{cases} \tag{18}$$

- *Interaction space cost "Obstacle":* The activities concerned by this function are direct activities or with interaction target, so the person must see the robot. We must check if there is no obstacle between the person and the robot that hides to one another, by using the $F_{EIObspij}$ function.

$$F_{EIObspij} = \begin{cases} \infty \text{ if } Obs \\ 0 \quad \text{else.} \end{cases} \tag{19}$$

We determined the parameters of the cost functions, according to the dimensions of the interaction space (Fig. 4).

F-Formation Type Cost: The people in a group interact together according to a spatial organization type, called F-Formation. In order to interact with a group, it is necessary to respect its F-Formation type (which is deducted from the person's location of the group), and the agent region changes according to the F-Formation type. Consequently, this function $F_{Formpij}$ includes three types of costs.

$$F_{Formpij} = \begin{cases} F_{Lpij} & \text{if } Type_F = \text{L-shape} \\ F_{Vispij} & \text{if } Type_F = \text{Vis-a-vis} \\ F_{Sidepij} & \text{if } Type_F = \text{Side by side} \end{cases} \tag{20}$$

- *L-shape:* The "L-shape" F-Formation is a spatial organization, where people set up themselves in the form of L. A third person who joins this group can keep the L-shape, or change it into an U-shape. In the case of a scientific mediator robot, which must be in the vision field of all the group persons, the U-shape is the most suitable.
  The best location is the point M_C by considering an error tolerance (Fig. 5). To determine the coordinates of this location, we will use this U-curve equation, where a represents the half-side of the square. The parameter K must be small to have a U-curve:

$$y = K\left(\frac{1}{(\sqrt{a^2 - x^2}} - \frac{1}{a}\right) \tag{21}$$

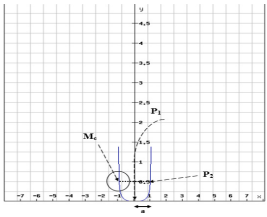


Fig. 5. Illustration of U-shape curve and positioning principle

- *Vis-a-vis:* The "Vis-a-vis" F-Formation is a spatial organization, where people come face to face. A third person who joins the group, must target a location at the P-space of this F-Formation, which is represented by the space between two circles, encompassing the locations of the group persons. The cost function F_{Vispij} must prefer all this P-space.
- *Side by side:* The "Side by side" F-Formation is a spatial organization, where people stand side by side. A third person who joins this group, can stand next to the two persons, or change this organization into an L-shape. In the case of a scientist mediator robot, which must be in the vision field of all the persons in the group, the L-shape is the most suitable.

The best location is to get in L-shape with the other persons. In order to cover the L-shape of each person in the group, we use the equation of U-curve as in the case of "L-shape" F-Formation (Eq. (21)), by covering both sides of the curves (Fig. 6).

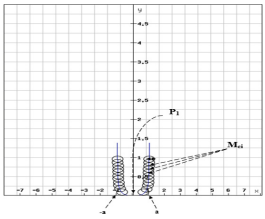


Fig. 6. Illustration of U-shape curve and positioning principle

Activity Space Cost

- *The activity space cost function* F_{Actpij}*:* This space includes all the space, objects, and targets of the activity. When an activity does not require moving, we will only check if the position of the person is an authorized position for the interaction. Whereas when the activity requires movement, the robot must put itself in the best position of interaction. If this activity has not an interaction target, then the F_{Actpij} function will only serve to limit the activity space $SAct$. In the opposite case, we will need more precision, so we vary this function according to the positive distance D_{Targij} between the cell M_{ij} and the interaction target, taking into account the absolute value of the orientation difference Ori_{Targij} between the target and the cell M_{ij}, and this in the case of an activity with a target. We define this cost function as follows:

$$F_{Actpij} = \begin{cases} 0 & \text{if } M_{ij} \in SAct \text{ and } \overline{Targ} \\ A_{pij} & \text{else if } M_{ij} \in SAct \\ \infty & \text{else.} \end{cases} \tag{22}$$

Where:

$$A_{pij} = Move(F_{ActDispij} + F_{ActOripij}) \tag{23}$$

"$F_{ActDispij}$" is the activity space cost function of the distance and "$F_{ActOripij}$" is the activity space cost function of the orientation.

* Activity space cost "Distance": The robot must not be too close or too far from the target, we use the function $F_{ActDispij}$. This function is a Gaussian function illustrated in Fig. 7, which depends on the absolute value of the distance D_{Targij} ($X = |D_{Targij}|$) between the target and the cell M_{ij}.

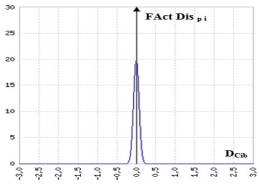


Fig. 7. Graph representating the activity space cost "Distance"

* Activity space cost "Orientation": The robot must not be in the perception space of the target, we use the function $F_{ActOripij}$. This function has the same form as the function $F_{ActDispij}$ (Fig. 7). Its values depend on the absolute value of the orientation difference O_{Targij} ($X = |O_{Targij}|$), between the target and the cell M_{ij}. The values of the $F_{ActOripij}$ function are inversely proportional to the O_{Targij} orientation.

We determined the cost functions parameters, according to the dimensions of the activity space.

- *The activity space SAct:* This space has a rectangular shape. Its center (O_{SAct}) and its dimensions (L_{SAct} and l_{SAct}) vary from one activity to another. So, we study the $SAct$ space of each activity.
  * "Welcome" activity: This activity involves standing in front of the door and inviting people to enter the room with only a hand sign. It is described by this model:
    "Welc" = Act (gesture, None, Visibility space, No, None, Door, Yes)
    The modality of this activity is the gesture, it is a priority direct activity (without support) without displacement, without object and with a target (the door). Its space is the visibility space. This space $SAct$ is all the space of the person to interact with, not too far or too close or next the door (Fig. 8). We set the center of the door in the middle of one side of the rectangle $SAct$. Its dimensions are for this activity: $L_{SAct} = l_{SAct} = 3m$. This space is shown in dashed line in Fig. 8.

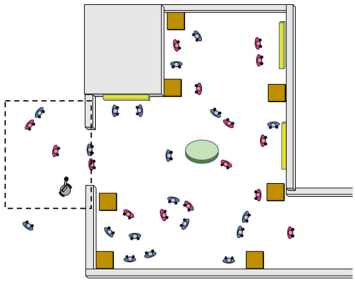


Fig. 8. Activity space of "Welcome" activity

* Activity "Speaker": This activity consists of inviting the persons to the conference room, it is described by this model:
"Spk" = Act(Speech, Speaker, Audibility Space, No, None, None, Yes)
The modality of this activity is speech, it is an indirect priority activity that uses a speaker but no displacement required neither object nor target. Its space is the audibility space. This $SAct$ space is the space where the person should be in order to interact with her, i.e. in the speaker audibility space (Fig. 9). We set the center of the rectangle O_{SAct} in the middle of the room. Its dimensions are for this activity equal to the room dimensions. This space is shown in dashed line in Fig. 9.

Fig. 9. Activity space of the "Speaker" activity

* Activity "Poster": This activity consists in explaining the posters to the guests, it is described by this model:
"ExP" = Act(Speech, None, Audibility Space, Yes, None, Poster, Yes)
The modality of this activity is the speech, it is a priority direct activity, which requires a displacement to reach the poster. Its target is the poster, but there is no object. Its space is the audibility space. This space $SAct$ is the space where the robot must be placed in order to interact with the person, i.e. neither too far nor too close nor behind the poster (Fig. 10). We fix the center of the poster in the middle of one side of the rectangle $SAct$. This space is shown in Fig. 10 in dashed line.

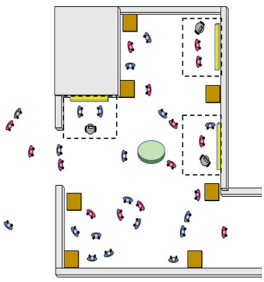


Fig. 10. Activity space of the "Poster" activity

* Activity "Serve Coffee": This activity consists in serving coffee to the guests, it is described by this model:
"Serv" = Act(Exchange, None, Exchange Space, Yes, Coffee, None, No)
The modality of this activity is the exchange, it is a direct secondary activity, which requires a displacement to reach the person. Its interaction object is the coffee, but it does not have a target. Its space is the exchange space. This space $SAct$ is the space where the robot must be placed in order to interact with the person, i.e. close to the person and not behind her (Fig. 11). We set the position of the person in the middle of one side of the rectangle $SAct$. This space is shown in dashed line in Fig. 11.

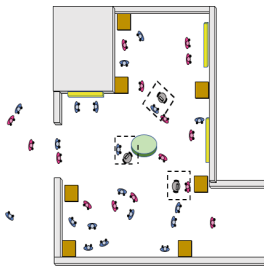


Fig. 11. Activity space of "Serve Coffee" activity

* Activity "Put Coffee": This activity consists in putting the coffee on a table, in the case where the robot can not reach the person. This activity is described by this model:
"Put" = Act (Exchange, Table, Exchange Space, Yes, Coffee, Table, Yes)
The modality of this activity is the exchange, it is a priority indirect activity, it requires a displacement to reach the target which is the table, its object is the coffee. Its space is the exchange space. This space $SAct$ is the space where the robot must be placed in order to interact with the person, ie close to the tables (Fig. 12). We set the center of the rectangle O_{SAct} in the center of the table. This space is shown in dashed line in Fig. 12.

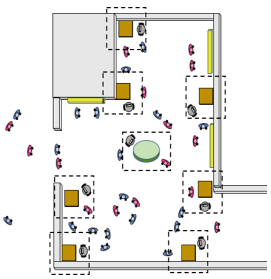


Fig. 12. Activity space of the "Put Coffee" activity

- *Set of activity spaces:* Some activities have multiple activity spaces, and the robot must choose the best space for this activity. For example, for the "Poster" activity, if we have several posters (Fig. 10), in order to interact with the person P, the robot must choose the activity space $SAct$, including the closest poster of this person, and which is in her perception space. Similar to the "Put coffee" activity, if there are several tables in the room (Fig. 12), the robot must choose the activity space $SAct$ containing the closest table to this person, and which is in her perception space, so she can see where the robot has dropped her coffee. We define the set of activity spaces for N_s space $SAct$:

$$SAct = \{SAct_0, SAct_2, .., SAct_q, ..., SAct_{N_s}\} \tag{24}$$

In order to select the best space $SAct_c$ of the target $Targ_c$ to interact with the person P, the robot must choose the space $SAct_q$ containing the interaction target $Targ_q$ which is accessible, closest to the person and in her vision field. We are looking for the corresponding target for this space, using our computed function F_{EIpij}, to find the target $Targ_q(x_q, y_q)$ that has the lowest cost.

$$Targ_c = \left\{ Targ_q \text{ if } \min_{q=0}^{N_s} F_{EIpx_qy_q} \text{ and } Accb_{Targ_q} = 1 \right. \tag{25}$$

The interaction cost function F_{Intij} represents the total weight of the cell M_{ij}, which is the weighted sum of all these studied costs.

Now, using this function, as well as the feasibility, the accessibility and the determination of Goal equations, we will select the most optimal Goal G, considering the modality and nature of the activity, as well as different criteria and social rules.

3.8 Implementation

This approach is tested by using the environment and the scenario described previously. Where, S is the start position and G is the goal position, I is the group to interact with and A is the group to avoid. The trajectory of the robot is represented by the black diamonds. We start by testing each activity separately for different types of F-Formation, different orientations and locations of persons. We evaluate results in simulation and on a real robot for the "Poster" and the "Put Coffee" activities.

– Simulation results:
We use the stage ROS simulator for evaluation. The environment of simulation is obtained by the map of our real environment (Fig. 13).

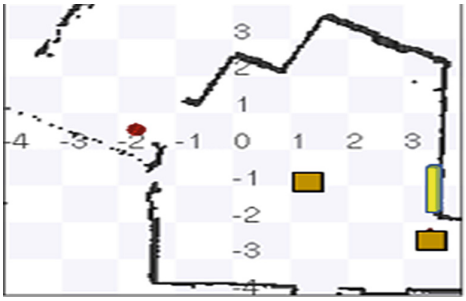
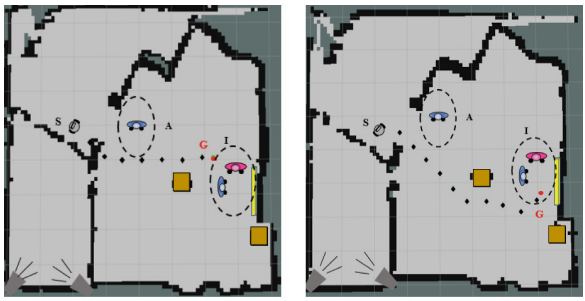


Fig. 13. Simulation environment of Stage ROS

- Activity "Poster"
  The simple navigation tests of this activity show that the robot takes the shortest path and the nearest location of interaction. This approach does not consider the modality and the nature of the activity, as well as security and social rules. The robot approaches the group to explain a poster without respecting the type of F-Formation of the group I, neither be in poster and group perception space. This is not the right way to explain a poster (Fig. 14a). While, the DPMA tests show that the trajectory of the robot avoids the personal space of the A person. The robot joins the I group to interact, by respecting the F-Formation types. It modifies the "L-Shape" group organisation to the "U-Shape". Then, it respects the space of poster, the space the group perception. Thus the feasibility of activity is satisfied (Fig. 14b).



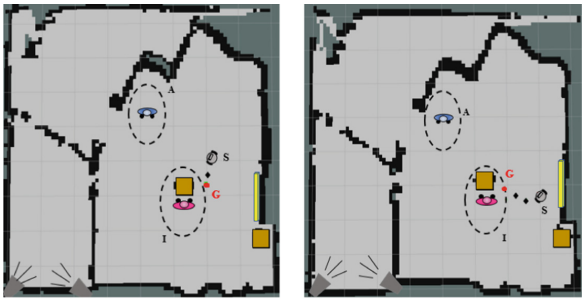
(a) (b)

Fig. 14. Results of "Poster" activity. a: Simple navigation. b: The DPMA

- Activity "Put Coffee"
  For this activity, the simple navigation and the DPMA tests provide the same results, because the nearest location of interaction from the robot given by the simple navigation is near the location with a low cost. But the

simple navigation considers the activity as an activity with a single target, by selecting the table of interaction (Fig. 15a). Our approach considers this activity as multi-target activity, ie the robot finds the right target, the closest to the person and in the space of person perception. The activity feasibility is checked, the robot selects the right table to interact (Fig. 15b).

– Experimentation on real robot:
We use the environment of Fig. 16a with the pioneer robot. We present the implementation results of the "Poster" and the "Put Coffee" activities, tested with several targets and spatial organisations.
  - Robotic platform:
    The Pioneer is an indoor robot (Fig. 16b). It is equipped with a laser scanner, and connected to a laptop by USB port. The laptop contains a robotic control software "ROS", the DPMA package and other drivers.



(a) (b)

Fig. 15. Results of "Put coffee" activity. a: Simple navigation. b: The DPMA

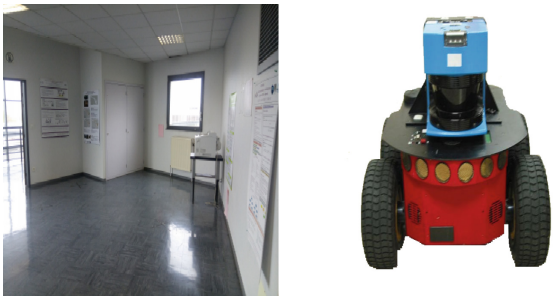


Fig. 16. The environment and the pioneer robot

- The "Poster" activity:
  In this activity, there is one person A to avoid, and the group I with a "L-shape" F-Formation to interact with (Fig. 17a, 17d). The result shows that the correct poster is selected. The person A is avoided by front. The I group is approached without crossing its personnel space. The interaction location is selected in the poster space. The "L-shape" formation is modified to a "U-shape", and the robot orients itself towards the group (Fig. 17c, 17f).
- The "Put Coffee" activity:
  The robot should put coffee on an accessible table to the I group, which is in "L-shape" F-Formation (Fig. 18a, 18d). This result shows that the robot chooses the nearest table to the I group and in the space of group perception, and the table selected is the correct target. When the person is from back, the robot avoids passing near her (Fig. 18b, 18e). The selected interaction location is in the exchange space and in the space of persons interaction. The robot orients itself towards the table and persons (Fig. 18c, 18f).
  We presented the most interesting activities results, to combine and verify several criteria. We obtain good results validating the "DPMA" approach.

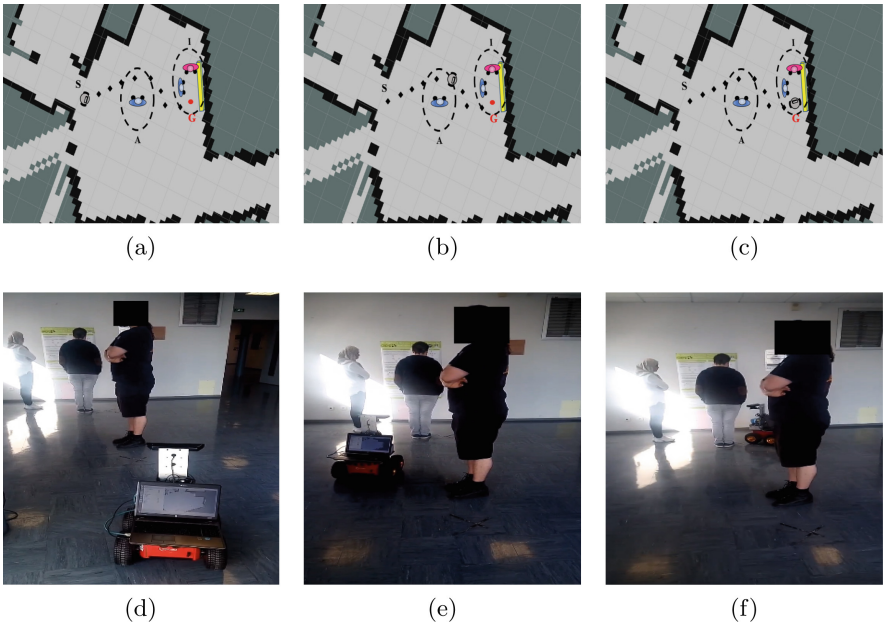


Fig. 17. Results of "Poster" activity. a, b, c: Robot trajectory. d, e, f: The real video

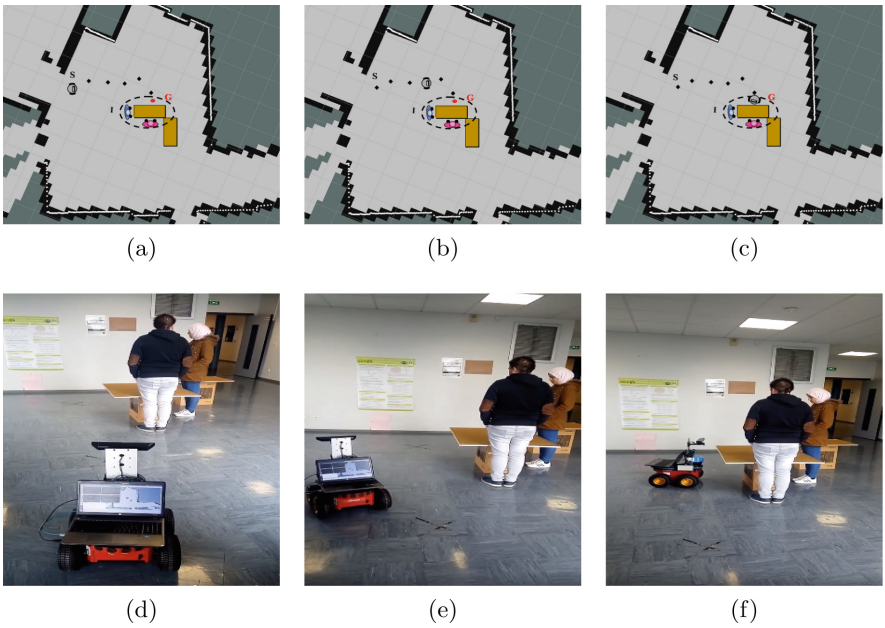


Fig. 18. Results of "Put Coffee" activity of experimentation 1. a, b, c: Robot trajectory. d, e, f: The real video

4 Conclusion

In this work, we present a formal framework of the dynamic proxemia modeling approach (DPMA), using the proposed model of activity and several criteria and social conventions. We implemented the "DPMA" in simulation using the "Stage" simulator and on a real robotic platform, providing promising results. In future works, we consider the dynamic environment and model the activity space. A validation with reception and guidance scenarios is suggested, in order to evaluate the DPMA according to robustness and performance criteria.

References

1. Hall, E.: The Hidden Dimension. Doubleday, vol. 6, no. 1 (1966)
2. Kendon, A.: Spacing and orientation in copresent interaction. In: Esposito, A., Campbell, N., Vogel, C., Hussain, A., Nijholt, A. (eds.) Second COST 2102. LNCS, vol. 5967, pp. 1–15. Springer, Heidelberg (2010)
3. Pandey, A.K., Alami, R.: A framework for adapting social conventions in a mobile robot motion in human-centered environment. In: 14th International Conference on Advanced Robotics, Munich, Germany (2009)
4. Dewantara, B.S.B., Miura, J.: Generation of a socially aware behavior of a guide robot using reinforcement learning. In: International Electronic Symposium (2016)

5. Ramirez, O.A.I., Khambhaita, H., Chatila, R., Chetouani, M., Alami, R.: Robots learning how and where to approach people. In: IEEE 25th International Symposium on Robot and Human Interactive Communication (RO-MAN) (2016)
6. Lindner, F.: A conceptual model of personal space for human-aware robot activity placement. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Congress Center Hamburg (2015)
7. Papadakis, P., Spalanzani, A., Laugier, C.: Social mapping of human-populated environments by implicit function learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2013)
8. Svenstrup, M., Tranberg, S., Andersen, H.J., Bak, T.: Pose estimation and adaptive robot behaviour for human-robot interaction. In: IEEE International Conference on Robotics and Automation Kobe International Conference Center, Japan (2009)
9. Charalampous, K., Kostavelis, I., Gasteratos, A.: Recent trends in social aware robot navigation: a survey. Robot. Auton. Syst. **93**, 85–104 (2017)
10. Vazquez, M., Steinfeld, A., Hudson, S.E.: Maintaining awareness of the focus of attention of a conversation: a robot-centric reinforcement learning approach. In: 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) (2016)
11. Chen, W., Zhang, T., Zou, Y.: Mobile robot path planning based on social interaction space in social environment. Int. J. Adv. Robot. Syst. **15**, 1–10 (2018)
12. Charalampous, K., Kostavelis, I., Gasteratos, A.: Robot navigation in large-scale social maps: an action recognition approach. Expert Syst. Appl. **66**, 261–273 (2016)
13. Truong, X.T., Ngo, T.D.: Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. Int. J. Soc. Robot. **8**, 663–664 (2016)
14. Bellarbi, A., Mouaddib, A., Achour, N., Ouadah, N.: Dynamic proxemia modeling approach (DPMA) for navigation and interaction with a group of persons. In: The 35th ACM/SIGAPP Symposium On Applied Computing (SAC), Czech (2020, to appear)
15. Dijkstra, E.W.: A note on two problems in connection with graphs. Numer. Math. **1**, 269–271 (1959)

Aspects Regarding the Elaboration of the Geometric, Kinematic and Organological Study of a Robotic Technological Product *"Humanitarian PetSim Robot"* Used as an Avant-Garde Element of the Human Factor in High Risk Areas

Silviu Mihai Petrişor[1(✉)] and Mihaela Simion[2]

[1] "Nicolae Bălcescu" Land Forces Academy, Revoluţiei Street, no. 3-5, 550170 Sibiu, Romania
robmilcap@gmail.com
[2] Technical University of Cluj-Napoca, Muncii Avenue, no. 103-105, 400641 Cluj-Napoca, Romania

Abstract. In this paper, the authors present a tracked mobile robot structure that is the subject of national invention patent number Ro a 00562 from 2017, granted by the State Office for Inventions and Trademarks, Bucharest, Romania, to our institution. For this specific tracked mobile robot structure, the authors present the mathematical algorithm in order to determinate the operational and generalized coordinates by using the method of 3 * 3 rotation matrices and the iterative method for a tracked mobile robot structure, respectively. A brief description of the organology of this technological product implemented in operations of humanitarian demining is given. Moreover, we present the working area chart of the tracked robot. The tracked mobile robot prototype, made in the Mechanical Engineering Laboratory II within our institution, pertains to the field of advanced military technologies and falls in the categories of artificial intelligence, technological humanism and human-artificial partnership. The paper highlights, in detail, the advantages, the novelty and the originality of the proposed solution, the technical problem solved by the technological robotic product, the operating mode and the use of the mobile tracked robot in special operations.

Keywords: Mobile tracked robot · Concern for the health of the planet · Protection of the human factor · Humanitarian demining

1 Short Description of the Technological Product

The invention refers to the creation of a technological product - tracked robot for humanitarian demining operations - *"Humanitarian PetSim Robot"*, integrated in the technical field - advanced military technologies, from the category of tracked mobile robots that are capable of replacing the human factor (as a vanguard element) in high-risk areas where

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 322–334, 2021.
https://doi.org/10.1007/978-3-030-55180-3_24

the health and life of a human operator are likely to be in danger. Our robot helps human personnel to avoid accidental detonation [6]. It also helps by detecting and demining anti-personnel and armor-piercing minefields in countries affected by military conflicts and thus it contributes to giving them back their economic and social circuit according to the standards stipulated by the UNO [1]. There are tracked robots that are currently being used in humanitarian demining operations, an example in this respect being the *Nemesis robot HD*, respectively the *RMA tEODor robot* [4, 5]. The disadvantages of these robots consist in the fact that they use fossil fuels for functioning. In this case, the running time is reduced, so there occurs the impossibility to simultaneously perform the operations of detecting and demining unexploded munitions thus increasing energy consumption. These robots have a complicated construction, with large dimensions, which leads to the diminution of flexibility and of the working space, respectively. Less attention is paid to modular interchangeability and, implicitly, to the possibility to carry out, in a timely manner, several operations specific to humanitarian demining.

The ever-changing military environment must be connected to the current imperatives of responsible intelligence and applied human technology in conjunction with the initiatives to ensure a stable balance between global operational challenges, protecting the human factor and taking care of the health of the planet by reducing pollution through the use of innovative renewable energy solutions.

The technical problem solved by our invention consists in increasing the flexibility of action through the simultaneous accomplishment of several operations, specific to humanitarian demining, combined with the reduction of pollution, minimization of energy consumption, modular exchange according to the requirements of the assigned missions, as well as the increase in the time allotted for the operations that are to be accomplished. The robot overcomes the disadvantages listed above and solves the technical problem by being capable to detect and demine, simultaneously, anti-personnel mines and anti-armor mines in minefields by means of high-performance detection equipment (video, audio, and radio communication), mounted on the mechanical structure of a translation system made up of nut-screw-guide elements, attached to the front of the tracked base and electrically driven, step by step, by electric motors. The technological product is able to move autonomously, by means of electric motors that take energy from solar photovoltaic cells encased in solar panels. It is provided with a storage compartment for the explosive necessary for humanitarian demining. It has a completely modularized compact structure, easy to mount and maintain, using materials and components resistant to dangerous environments, the communication between the human operator and the robot taking place wirelessly, based on a predefined computer program (Fig. 1).

The tracked robot is composed of two main structures: the tracked base and the serial-modular robot of TRTTR (3 Translations and two Rotations) type (Fig. 1 and Fig. 2) with the translation system of the device for unexploded mine detection.

We should also specify that, on the one hand, each mobile crew of the robot operates independently, and, in the event of failure of an electric drive motor, this can be replaced without affecting the operation of the other modules. The repair time is short and it does not compromise the robot's operations during this period. On the other hand, the translation or rotation module and the horizontal and vertical arms respectively, can be mounted

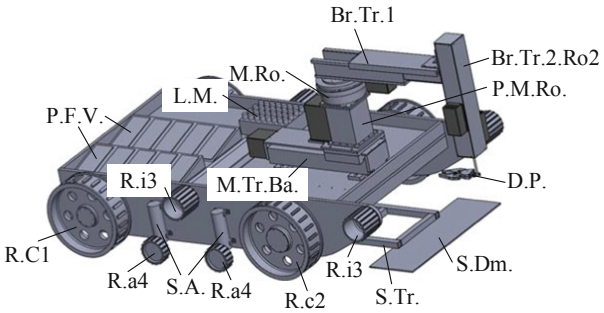


Fig. 1. The 3D model of the tracked robot

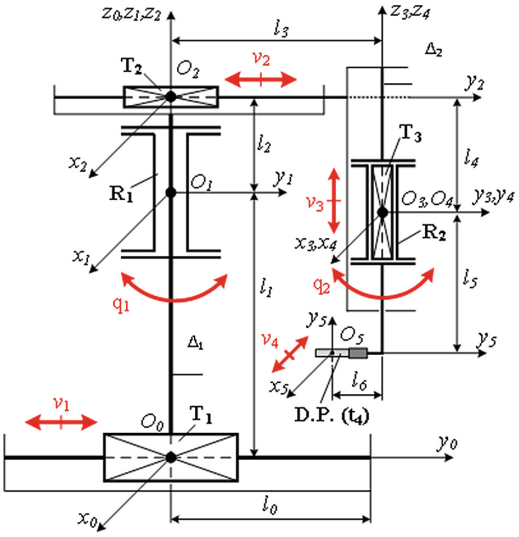


Fig. 2. The kinematic diagram of the TRTTR modular serial tracked robot

on various architectures of robotic structures necessary for humanitarian demining. Extra translation and/or rotation modules can also be mounted for the accomplishment of the optional tasks/operations assigned to the robot.

2 The Geometric Model of the Technological Product

The kinematic diagram of the TRTTR modular serial tracked robot is presented in Fig. 2 where the following notations are introduced: l_i, $i = 0 \div 6 \rightarrow$ constructive robot parameters, q_k, $k = 1 \div 2$; v_k, $k = 1 \div 3 \rightarrow$ generalized robot coordinates.

The robot is composed of: horizontal translation module 1 at the robotic base (T_1), rotation module 2 along the vertical axis in the robotic arm (R_1), horizontal translation from the component arm module 3 (T_2), vertical translation module 4 in the robotic

arm structure (T_3) and rotation module 5 along the vertical axis in the robotic arm (R_2) assembled together with the clamping device (PD).

First, direct and inverse modeling will be carried out, using the 3 * 3 rotation matrices method and the algebraic method, respectively [2].

The relative orientation matrices expressing the orientation of each O_i, $(i = 0 \div 5)$ system, in relation to the previous T_{i-1}, system are the following:

$$[R]_1^0 = [R]_3^2 = [R]_4^3 = [R]_6^5 = [I_3] = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}; [R]_2^1 = R(\bar{k}, q_1) = \begin{vmatrix} cq_1 & -sq_1 & 0 \\ sq_1 & cq_1 & 0 \\ 0 & 0 & 1 \end{vmatrix};$$

$$[R]_5^4 = R(\bar{k}, q_2) = \begin{vmatrix} cq_2 & -sq_2 & 0 \\ sq_2 & cq_2 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \quad (1)$$

The position vectors relative to the O_i,$(i = 0 \div 5)$ origins of T_i,$(i = 1 \div 5)$ systems in relation to the previous system have the following expressions:

$$\bar{r}_1^0 = \begin{vmatrix} 0 \\ l_0 + v_1 \\ 0 \end{vmatrix}; \bar{r}_2^1 = \begin{vmatrix} 0 \\ 0 \\ l_1 \end{vmatrix}; \bar{r}_3^2 = \begin{vmatrix} 0 \\ l_3 + v_2 \\ l_2 \end{vmatrix}; \bar{r}_4^3 = \begin{vmatrix} 0 \\ 0 \\ l_4 + v_3 \end{vmatrix}; \bar{r}_5^4 = \begin{vmatrix} 0 \\ 0 \\ l_5 \end{vmatrix}; \bar{r}_6^5 = \begin{vmatrix} 0 \\ l_6 \\ 0 \end{vmatrix}. \quad (2)$$

In order to determine the set of $(\alpha_z - \beta_x - \gamma_z)$ independent orientation parameters, the following matrix relation must be considered:

$$[R]_6^0 = R(\alpha_z - \beta_x - \gamma_z), \quad (3)$$

where:

$$R(\alpha_z - \beta_x - \gamma_z) = \begin{vmatrix} c\alpha_z c\gamma_z - s\alpha_z s\gamma_z c\beta_x & -c\alpha_z s\gamma_z - s\alpha_z c\gamma_z c\beta_x & s\alpha_z s\beta_x \\ s\alpha_z c\gamma_z + c\alpha_z s\gamma_z c\beta_x & -s\alpha_z s\gamma_z + c\alpha_z c\gamma_z c\beta_x & -c\alpha_z s\beta_x \\ s\gamma_z s\beta_x & c\gamma_z s\beta_x & c\beta_x \end{vmatrix}. \quad (4)$$

The absolute rotation matrices, which express the orientation of each O_i, $(i = 2 \div 6)$ system in relation to the O_0 fixed system, are the following:

$$[R]_2^0 = [R]_1^0 \cdot [R]_2^1 = \begin{vmatrix} cq_1 & -sq_1 & 0 \\ sq_1 & cq_1 & 0 \\ 0 & 0 & 1 \end{vmatrix}; [R]_3^0 = [R]_2^0 \cdot [R]_3^2 = \begin{vmatrix} cq_1 & -sq_1 & 0 \\ sq_1 & cq_1 & 0 \\ 0 & 0 & 1 \end{vmatrix};$$

$$[R]_4^0 = [R]_3^0 \cdot [R]_4^3 = \begin{vmatrix} cq_1 & -sq_1 & 0 \\ sq_1 & cq_1 & 0 \\ 0 & 0 & 1 \end{vmatrix}; [R]_5^0 = [R]_4^0 \cdot [R]_5^4 = \begin{vmatrix} cq_1cq_2 - sq_1sq_2 & -cq_1sq_2 - sq_1cq_2 & 0 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 \\ 0 & 0 & 1 \end{vmatrix};$$

$$[R]_6^0 = [R]_5^0 \cdot [R]_6^5 = \begin{vmatrix} cq_1cq_2 - sq_1sq_2 & -cq_1sq_2 - sq_1cq_2 & 0 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \quad (5)$$

Given relation (4) and the last relation in set (5), the $(\alpha_z - \beta_x - \gamma_z)$ independent orientation parameters set is determined:

$$\begin{vmatrix} \alpha_z \\ \beta_x \\ \gamma_z \end{vmatrix} = \begin{vmatrix} -\frac{\pi}{2} \\ \frac{\pi}{2} + q_1 \\ q_2 \end{vmatrix}. \tag{6}$$

The following relations are used to determine the origin position of each O_i, $(i = 1 \div 6)$ reference system in relation to the O_{i-1} system:

$$\bar{p}_{10} = \bar{r}_1^0 = \begin{vmatrix} 0 \\ l_0 + v_1 \\ 0 \end{vmatrix}; \bar{p}_{21} = [R]_1^0 \cdot \bar{r}_2^1 = \begin{vmatrix} 0 \\ 0 \\ l_1 \end{vmatrix};$$

$$\bar{p}_{32} = [R]_2^0 \cdot \bar{r}_3^2 = \begin{vmatrix} -sq_1(l_3 + v_2) \\ cq_1(l_3 + v_2) \\ l_2 \end{vmatrix};$$

$$\bar{p}_{43} = [R]_3^0 \cdot \bar{r}_4^3 = \begin{vmatrix} 0 \\ 0 \\ l_4 + v_3 \end{vmatrix}; \bar{p}_{54} = [R]_4^0 \cdot \bar{r}_5^4 = \begin{vmatrix} 0 \\ 0 \\ l_5 \end{vmatrix};$$

$$\bar{p}_{65} = [R]_5^0 \cdot \bar{r}_6^5 = \begin{vmatrix} -(cq_1 sq_2 + sq_1 cq_2) l_6 \\ (-sq_1 sq_2 + cq_1 cq_2) l_6 \\ 0 \end{vmatrix}. \tag{7}$$

Given the relations (7), we use the following expressions to obtain the position of the origin of every O_i system in relation to the O_0 system fixed to the robotic base:

$$\bar{p}_1 = \bar{p}_{10} = \begin{vmatrix} 0 \\ l_0 + v_1 \\ 0 \end{vmatrix}; \bar{p}_2 = \bar{p}_1 + \bar{p}_{21} = \begin{vmatrix} 0 \\ l_0 + v_1 \\ l_1 \end{vmatrix};$$

$$\bar{p}_3 = \bar{p}_2 + \bar{p}_{32} = \begin{vmatrix} -sq_1(l_3 + v_2) \\ l_0 + v_1 + cq_1(l_3 + v_2) \\ l_1 + l_2 \end{vmatrix};$$

$$\bar{p}_4 = \bar{p}_3 + \bar{p}_{43} = \begin{vmatrix} -sq_1(l_3 + v_2) \\ l_0 + v_1 + cq_1(l_3 + v_2) \\ l_1 + l_2 + l_4 + v_3 \end{vmatrix}; \bar{p}_5 = \bar{p}_4 + \bar{p}_{54} = \begin{vmatrix} -sq_1(l_3 + v_2) \\ l_0 + v_1 + cq_1(l_3 + v_2) \\ l_1 + l_2 + l_4 + l_5 + v_3 \end{vmatrix};$$

$$\bar{p}_6 = \bar{p}_5 + \bar{p}_{65} = \begin{vmatrix} -sq_1(l_3 + v_2) - (cq_1 sq_2 + sq_1 cq_2) l_6 \\ l_0 + v_1 + cq_1(l_3 + v_2) + (cq_1 cq_2 - sq_1 sq_2) l_6 \\ l_1 + l_2 + l_4 + l_5 + v_3 \end{vmatrix}. \tag{8}$$

Next we introduced the column vector in the operational coordinates:

$$\bar{X}^0 = \left[p_x\, p_y\, p_z\, \alpha_z\, \beta_x\, \gamma_z \right]^T = \left[f_j(q_k, v_k, k = 1 \div h), j = 1 \div 6 \right]^T, \tag{9}$$

The results obtained from relation (6) and the last relation in set (8), helped us to obtain the following expression:

$$\bar{X}^0 = \begin{vmatrix} p_{x6} \\ p_{y6} \\ p_{z6} \\ \dots \\ \alpha_z \\ \beta_x \\ \gamma_z \end{vmatrix} = \begin{vmatrix} -sq_1(l_3 + v_2) - (cq_1sq_2 + sq_1cq_2)l_6 \\ l_0 + v_1 + cq_1(l_3 + v_2) + (cq_1cq_2 - sq_1sq_2) \\ l_1 + l_2 + l_4 + l_5 + v_3 \\ \dots \\ -\pi/2 \\ \pi/2 + q_1 \\ q_2 \end{vmatrix}.. \tag{10}$$

The homogenous transformation matrix is the following:

$$[T]_6^0 = \begin{vmatrix} [R]_6^0 \vdots [P]^6 \\ \dots \vdots \dots \\ 0\ 0\ 0 \vdots 1 \end{vmatrix}$$

$$= \begin{vmatrix} cq_1cq_2 - sq_1sq_2 & -cq_1sq_2 - sq_1cq_2 & 0 \vdots & -sq_1(l_3 + v_2) - (cq_1sq_2 + sq_1cq_2)l_6 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 \vdots & l_0 + v_1 + cq_1(l_3 + v_2) + (cq_1cq_2 - sq_1sq_2)l_6 \\ 0 & 0 & 1 \vdots & l_1 + l_2 + l_4 + l_5 + v_3 \\ \dots & \dots & \dots \vdots & \dots \\ 0 & 0 & 0 \vdots & 1 \end{vmatrix}. \tag{11}$$

Relation (11) expresses the equations of the direct geometric model for the TRTTR-type serial modular tracked robot. By means of these equations we can determine the position and orientation of the clamping device of the robot.

Starting from the matrix expressing the position and orientation of the clamping device of the robot in relation to the O_0 fixed system, we obtained the following expression:

$$[T]_6^0 = \begin{vmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{vmatrix}, \tag{12}$$

After the identification of the homogeneous transformation matrix (11) elements, the equations of the inverse geometric model or the geometric command functions were derived:

$$\begin{vmatrix} v_1 \\ q_1 \\ v_2 \\ q_2 \\ v_3 \end{vmatrix} = \begin{vmatrix} p_y - \left[l_0 + q_1(l_3 + v_2) + r_{11}l_6\right] \\ a\tan\left[\frac{-(p_x + r_{12}l_6)}{p_y - l_0 - v_1 - r_{11}l_6}\right] \\ \pm\sqrt{\left(p_x + r_{12}l_6 - sq_1 l_3\right)^2 + \left[p_y - \left(l_0 + v_1 + r_{11}l_6 + cq_1 l_3\right)\right]^2} \\ a\tan 2(-r_{12}, r_{22}) \\ p_z - (l_1 + l_2 + l_4 + l_5) \end{vmatrix}. \tag{13}$$

3 The Kinematic Model of the Technological Product

The direct kinematic modeling [3] originates in the geometric modeling that helped us to determine the homogenous transformation matrices (see Fig. 1 and 3):

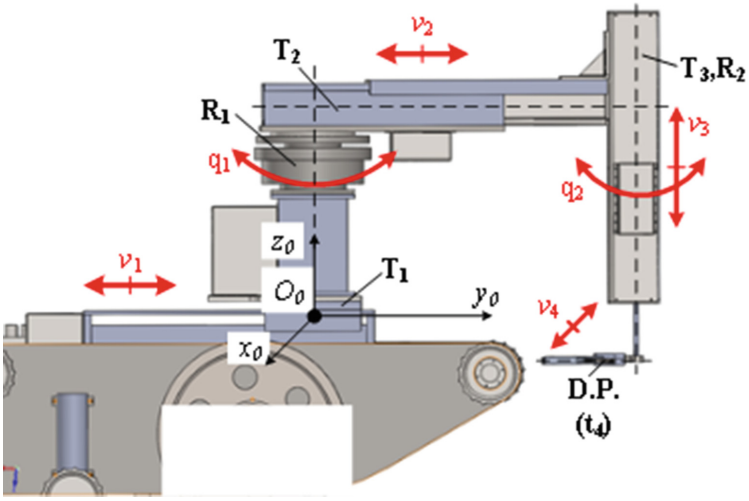


Fig. 3. The 3D model of the TRTTR robot

$$[T]_1^0(t) = \left|\begin{array}{cccc:c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_0 + v_1 \\ 0 & 0 & 1 & 0 \\ \hdashline \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 \end{array}\right|; \quad [T]_2^1(t) = \left|\begin{array}{cccc:c} cq_1 & -sq_1 & 0 & 0 \\ sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ \hdashline \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 \end{array}\right|; \tag{14}$$

$$[T]_3^2(t)=\left|\begin{array}{cccc} 1 & 0 & 0 & -sq_1(l_3+v_2)\\ 0 & 1 & 0 & cq_1(l_3+v_2)\\ 0 & 0 & 1 & l_2\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|;\ [T]_4^3(t)=\left|\begin{array}{cccc} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & l_4+v_3\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|;$$

$$[T]_5^4(t)=\left|\begin{array}{cccc} cq_2 & -sq_2 & 0 & 0\\ sq_2 & cq_2 & 0 & 0\\ 0 & 0 & 1 & l_5\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|;\ [T]_6^5(t)=\left|\begin{array}{cccc} 1 & 0 & 0 & (-cq_1sq_2-sq_1cq_2)l_6\\ 0 & 1 & 0 & (-sq_1sq_2+cq_1cq_2)l_6\\ 0 & 0 & 1 & 0\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|,$$

We used the following respective matrices:

$$[T]_2^0(t)=\left|\begin{array}{cccc} cq_1 & -sq_1 & 0 & 0\\ sq_1 & cq_1 & 0 & l_0+v_1\\ 0 & 0 & 1 & l_1\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|;\ [T]_3^0(t)=\left|\begin{array}{cccc} cq_1 & -sq_1 & 0 & -sq_1(l_3+v_2)\\ sq_1 & cq_1 & 0 & l_0+v_1+cq_1(l_3+v_2)\\ 0 & 0 & 1 & l_1+l_2\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|;$$

$$[T]_4^0(t)=\left|\begin{array}{cccc} cq_1 & -sq_1 & 0 & -sq_1(l_3+v_2)\\ sq_1 & cq_1 & 0 & l_0+v_1+cq_1(l_3+v_2)\\ 0 & 0 & 1 & l_1+l_2+l_4+v_3\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|;$$

$$[T]_5^0(t)=\left|\begin{array}{cccc} cq_1cq_2-sq_1sq_2 & -cq_1sq_2-sq_1cq_2 & 0 & -sq_1(l_3+v_2)\\ sq_1cq_2+cq_1sq_2 & -sq_1sq_2+cq_1cq_2 & 0 & l_0+v_1+cq_1(l_3+v_2)\\ 0 & 0 & 1 & l_1+l_2+l_4+l_5+v_3\\ \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & 1 \end{array}\right|; \qquad (15)$$

$$[T]_6^0(t) = \left|\begin{array}{ccc:c} cq_1cq_2 - sq_1sq_2 & -cq_1sq_2 - sq_1cq_2 & 0 & -sq_1(l_3 + v_2) - (cq_1sq_2 + sq_1cq_2)l_6 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 & l_0 + v_1 + cq_1(l_3 + v_2) + (cq_1cq_2 - sq_1sq_2)l_6 \\ 0 & 0 & 1 & l_1 + l_2 + l_4 + l_5 + v_3 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 \end{array}\right|.$$

The inverse rotation matrices are determined by means of the following relation:

$$[R]_{i-1}^{i} = [[R]_i^{i-1}]^{-1} = [[R]_i^{i-1}]^T. \tag{16}$$

Thus,

$$\begin{gathered}[R]_0^1 = [R]_2^3 = [R]_3^4 = [R]_5^6 = [I_3] = \left|\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right|; [R]_1^2 = \left|\begin{array}{ccc} cq_1 & sq_1 & 0 \\ -sq_1 & cq_1 & 0 \\ 0 & 0 & 1 \end{array}\right|; \\ [R]_4^5 = \left|\begin{array}{ccc} cq_2 & sq_2 & 0 \\ -sq_2 & cq_2 & 0 \\ 0 & 0 & 1 \end{array}\right|.\end{gathered} \tag{17}$$

The unit vectors along the axes have the following expressions:

$$\bar{k}_1^1 = \left|\begin{array}{c} 0 \\ 1 \\ 0 \end{array}\right|; \bar{k}_2^2 = \left|\begin{array}{c} 0 \\ 0 \\ 1 \end{array}\right|; \bar{k}_3^3 = \left|\begin{array}{c} 0 \\ 1 \\ 0 \end{array}\right|; \bar{k}_4^4 = \left|\begin{array}{c} 0 \\ 0 \\ 1 \end{array}\right|; \bar{k}_5^5 = \left|\begin{array}{c} 0 \\ 0 \\ 1 \end{array}\right|. \tag{18}$$

The elements corresponding to the base of the robot are the following:

$$\bar{\omega}_0^0 = \left|\begin{array}{c} 0 \\ 0 \\ 0 \end{array}\right|; \bar{v}_0^0 = \left|\begin{array}{c} 0 \\ 0 \\ 0 \end{array}\right|; \bar{\varepsilon}_0^0 = \left|\begin{array}{c} 0 \\ 0 \\ 0 \end{array}\right|; \bar{a}_0^0 = \left|\begin{array}{c} 0 \\ 0 \\ g \end{array}\right|. \tag{19}$$

The operational angular velocities are the following:

$$\begin{gathered}\bar{\omega}_1^1 = [R]_0^1 \cdot \bar{\omega}_0^0 = \left|\begin{array}{c} 0 \\ 0 \\ 0 \end{array}\right|; \bar{\omega}_2^2 = [R]_1^2 \cdot \bar{\omega}_1^1 + \dot{q}_2 \times \bar{k}_2^2 = \left|\begin{array}{c} 0 \\ 0 \\ \dot{q}_2 \end{array}\right|; \bar{\omega}_3^3 = [R]_2^3 \cdot \omega_2^2 = \left|\begin{array}{c} 0 \\ 0 \\ \dot{q}_2 \end{array}\right|; \\ \bar{\omega}_4^4 = [R]_3^4 \cdot \bar{\omega}_3^3 = \left|\begin{array}{c} 0 \\ 0 \\ \dot{q}_2 \end{array}\right|; \\ \bar{\omega}_5^5 = [R]_4^5 \cdot \bar{\omega}_4^4 + \dot{q}_5 \times \bar{k}_5^5 = \left|\begin{array}{c} 0 \\ 0 \\ \dot{q}_2 + \dot{q}_5 \end{array}\right|; \bar{\omega}_6^6 = [R]_5^6 \cdot \bar{\omega}_5^5 = \left|\begin{array}{c} 0 \\ 0 \\ \dot{q}_2 + \dot{q}_5 \end{array}\right|.\end{gathered} \tag{20}$$

The operational linear velocities are the following:

$$\bar{v}_1^1 = [R]_0^1 \cdot \left\{\bar{v}_0^0 + \bar{\omega}_0^0 \times \bar{r}_1^0\right\} + \dot{q}_1\bar{k}_1^1 = \begin{vmatrix} 0 \\ \dot{q}_1 \\ 0 \end{vmatrix}; \bar{v}_2^2 = [R]_1^2 \cdot \left\{\bar{v}_1^1 + \bar{\omega}_1^1 \times \bar{r}_2^1\right\} = \begin{vmatrix} sq_1\dot{q}_1 \\ cq_1\dot{q}_1 \\ 0 \end{vmatrix};$$

$$\bar{v}_3^3 = [R]_2^3 \cdot \left\{\bar{v}_2^2 + \bar{\omega}_2^2 \times \bar{r}_3^2\right\} + \dot{q}_3\bar{k}_3^3 = [I_3] \cdot \left\{\left\{\begin{vmatrix} sq_1\dot{q}_1 \\ cq_1\dot{q}_1 \\ 0 \end{vmatrix} + \begin{vmatrix} 0 & -\dot{q}_2 & 0 \\ \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}\right\} \times \begin{vmatrix} 0 \\ l_3 + v_2 \\ l_2 \end{vmatrix}\right\} + \dot{q}_3 \cdot \begin{vmatrix} 0 \\ 1 \\ 0 \end{vmatrix} =$$

$$= \begin{vmatrix} sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2) \\ cq_1\dot{q}_1 + \dot{q}_3 \\ 0 \end{vmatrix};$$

$$\bar{v}_4^4 = [R]_3^4 \times \left\{\bar{v}_3^3 + \bar{\omega}_3^3 \times \bar{r}_4^3\right\} + \dot{q}_4\bar{k}_4^4 = \begin{vmatrix} sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2) \\ cq_1\dot{q}_1 + \dot{q}_3 \\ \dot{q}_4 \end{vmatrix};$$

$$\bar{v}_5^5 = [R]_4^5 \times \left\{\bar{v}_4^4 + \bar{\omega}_4^4 \times \bar{r}_5^4\right\} = \begin{vmatrix} cq_2\left[sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2)\right] + sq_2\left(cq_1\dot{q}_1 + \dot{q}_3\right) \\ -sq_2\left[sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2)\right] + cq_2\left(cq_1\dot{q}_1 + \dot{q}_3\right) \\ \dot{q}_4 \end{vmatrix};$$

$$\bar{v}_6^6 = [R]_5^6 \times \left\{\bar{v}_5^5 + \bar{\omega}_5^5 \times \bar{r}_6^5\right\} = \begin{vmatrix} cq_2\left[sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2)\right] + sq_2\left(cq_1\dot{q}_1 + \dot{q}_3\right) - l_6\left(\dot{q}_2 + \dot{q}_5\right) \\ -sq_2\left[sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2)\right] + cq_2\left(cq_1\dot{q}_1 + \dot{q}_3\right) \\ \dot{q}_4 \end{vmatrix}. \tag{21}$$

The angular/operational accelerations can be expressed by using the relations:

$$\bar{\varepsilon}_1^1 = [R]_0^1 \times \bar{\varepsilon}_0^0 = \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}; \bar{\varepsilon}_2^2 = [R]_1^2 \times \bar{\varepsilon}_1^1 + \left\{[R]_1^2 \times \bar{\omega}_1^1 \times \dot{q}_2\bar{k}_2^2 + \ddot{q}_2\bar{k}_2^2\right\} = \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_2 \end{vmatrix};$$

$$\bar{\varepsilon}_3^3 = [R]_2^3 \times \bar{\varepsilon}_2^2 = \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_2 \end{vmatrix};$$

$$\bar{\varepsilon}_4^4 = [R]_3^4 \times \bar{\varepsilon}_3^3 = \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_2 \end{vmatrix};$$

$$\bar{\varepsilon}_5^5 = [R]_4^5 \times \bar{\varepsilon}_4^4 + \left\{[R]_4^5 \times \bar{\omega}_4^4 \times \dot{q}_5\bar{k}_5^5 + \ddot{q}_5\bar{k}_5^5\right\} = \begin{vmatrix} cq_2 & sq_2 & 0 \\ -sq_2 & cq_2 & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_2 \end{vmatrix} s$$

$$+\left\{\begin{vmatrix} cq_2 & sq_2 & 0 \\ -sq_2 & cq_2 & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 0 & -q_2 & 0 \\ \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_5 \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_5 \end{vmatrix}\right\} = \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_2 + \ddot{q}_5 \end{vmatrix};$$

$$\bar{\varepsilon}_6^6 = [R]_5^6 \times \bar{\varepsilon}_5^5 = \begin{vmatrix} 0 \\ 0 \\ \ddot{q}_2 + \ddot{q}_5 \end{vmatrix}. \tag{22}$$

$$\bar{a}_1^1 = [R]_0^1 \times \left\{\bar{a}_0^0 + \bar{\varepsilon}_0^0 \times \bar{r}_1^0 + \bar{\omega}_0^0 \times \left(\bar{\omega}_0^0 \times \bar{r}_1^0\right)\right\} + \left\{2\bar{\omega}_1^1 \times \dot{q}_1 \bar{k}_1^1 + \ddot{q}_1 \bar{k}_1^1\right\} = \begin{vmatrix} 0 \\ \ddot{q}_1 \\ g \end{vmatrix};$$

$$\bar{a}_2^2 = [R]_1^2 \times \left\{\bar{a}_1^1 + \bar{\varepsilon}_1^1 \times \bar{r}_2^1 + \bar{\omega}_1^1 \times \left(\bar{\omega}_1^1 \times \bar{r}_1^2\right)\right\} = \begin{vmatrix} sq_1\ddot{q}_1 \\ cq_1\ddot{q}_1 \\ g \end{vmatrix};$$

$$\bar{a}_3^3 = [R]_2^3 \times \left\{\bar{a}_2^2 + \bar{\varepsilon}_2^2 \times \bar{r}_3^2 + \bar{\omega}_2^2 \times \left(\bar{\omega}_2^2 \times \bar{r}_3^2\right)\right\} + \left\{2\bar{\omega}_3^3 \times \dot{q}_3 \bar{k}_3^3 + \ddot{q}_3 \bar{k}_3^3\right\} =$$

$$= [I_3] \times \{ \begin{vmatrix} sq_1\ddot{q}_1 \\ cq_1\ddot{q}_1 \\ g \end{vmatrix} + \begin{vmatrix} 0 & -\ddot{q}_2 & 0 \\ \ddot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} 0 \\ l_3 + v_2 \\ l_2 \end{vmatrix} + \begin{vmatrix} 0 & -\dot{q}_2 & 0 \\ \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} 0 & -\dot{q}_2 & 0 \\ \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} 0 \\ l_3 + v_2 \\ l_2 \end{vmatrix} \} +$$

$$+ \{2 \begin{vmatrix} 0 & -\dot{q}_2 & 0 \\ \dot{q}_2 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} 0 \\ \dot{q}_3 \\ 0 \end{vmatrix} + \begin{vmatrix} 0 \\ \ddot{q}_3 \\ 0 \end{vmatrix} \} = \begin{vmatrix} sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3 \\ cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3 \\ g \end{vmatrix};$$

$$\bar{a}_4^4 = [R]_3^4 \left\{\bar{a}_3^3 + \bar{\varepsilon}_3^3 \times \bar{r}_4^3 + \bar{\omega}_3^3 \times \left(\bar{\omega}_3^3 \times \bar{r}_4^3\right)\right\} + \left\{2\bar{\omega}_4^4 \times \dot{q}_4 \bar{k}_4^4 + \ddot{q}_4 \bar{k}_4^4\right\} =$$

$$= \begin{vmatrix} sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3 \\ cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3 \\ g + \ddot{q}_4 \end{vmatrix}; \bar{a}_5^5 = [R]_4^5 \left\{\bar{a}_4^4 + \bar{\varepsilon}_4^4 \times \bar{r}_5^4 + \bar{\omega}_4^4 \times \left(\bar{\omega}_4^4 \times \bar{r}_5^4\right)\right\} =$$

$$= \begin{vmatrix} cq_2\left[sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3\right] + sq_2\left[cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3\right] \\ -sq_2\left[sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3\right] + cq_2\left[cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3\right] \\ g + \ddot{q}_4 \end{vmatrix}; \qquad (23)$$

$$\bar{a}_6^6 = [R]_5^6 \left\{\bar{a}_5^5 + \bar{\varepsilon}_5^5 \times \bar{r}_6^5 + \bar{\omega}_5^5 \times \left(\bar{\omega}_5^5 \times \bar{r}_6^5\right)\right\} =$$

$$= \begin{vmatrix} cq_2\left[sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3\right] + sq_2\left[cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3\right] - l_6(\ddot{q}_2 + \ddot{q}_5) \\ -sq_2\left[sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3\right] + cq_2\left[cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3\right] - l_6(\dot{q}_2 + \dot{q}_5)^2 \\ g + \ddot{q}_4 \end{vmatrix}.$$

The operational kinematic parameters in system (T_6) can be written in the following form:

$$\dot{\bar{X}}^6 = \begin{vmatrix} cq_2\left[sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2)\right] + sq_2\left(cq_1\dot{q}_1 + \dot{q}_3\right) - l_6(\dot{q}_2 + \dot{q}_5) \\ -sq_2\left[sq_1\dot{q}_1 - \dot{q}_2(l_3 + v_2)\right] + cq_2\left(cq_1\dot{q}_1 + \dot{q}_3\right) \\ \dot{q}_4 \\ 0 \\ 0 \\ \dot{q}_2 + \dot{q}_5 \end{vmatrix}, \qquad (24)$$

$$\ddot{\bar{X}}^6 = \begin{vmatrix} cq_2\left[sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3\right] + sq_2\left[cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3\right] - l_6(\ddot{q}_2 + \ddot{q}_5) \\ -sq_2\left[sq_1\ddot{q}_1 - \ddot{q}_2(l_3 + v_2) - 2\dot{q}_2\dot{q}_3\right] + cq_2\left[cq_1\ddot{q}_1 - \dot{q}_2^2(l_3 + v_2) + 2\ddot{q}_3\right] - l_6(\dot{q}_2 + \dot{q}_5)^2 \\ g + \ddot{q}_4 \\ 0 \\ 0 \\ \ddot{q}_2 + \ddot{q}_5 \end{vmatrix}.$$

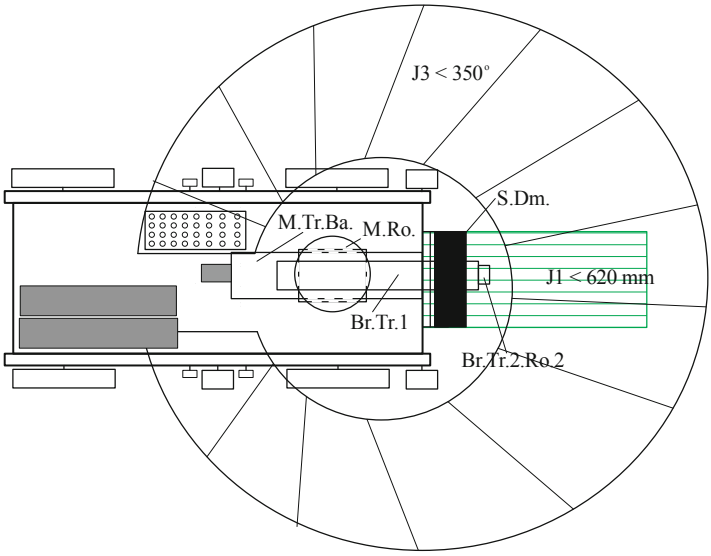


Fig. 4. The workspace defined by the kinematic axis J3 of the rotation module and of the translation system

The operational kinematic parameters in the fixed system (T_0) at the robot base are determined by using transformation relations. Thus we obtained:

$$\bar{v}_6^0 = [R]_6^0 \times \bar{v}_6^6; \bar{\omega}_6^0 = [R]_6^0 \times \bar{\omega}_6^6; \bar{a}_6^0 = [R]_6^0 \times \bar{a}_6^6; \bar{\varepsilon}_6^0 = [R]_6^0 \times \bar{\varepsilon}_6^6. \tag{25}$$

Operational velocity and acceleration in the fixed system (T_0) can be expressed by means of the relation below and the workspace of the robot is presented in Fig. 4:

$$\dot{\bar{X}}^0 = \begin{vmatrix} \bar{v}_6^0 \\ \bar{\omega}_6^0 \end{vmatrix}; \ddot{\bar{X}}^0 = \begin{vmatrix} \bar{a}_6^0 \\ \bar{\varepsilon}_6^0 \end{vmatrix}. \tag{26}$$

4 Applications

The technological product the invention refers to has applicability in (a) the military domain, by improving the operational flexibility in humanitarian operations of detecting and demining antipersonnel and anti-armor mines in dangerous areas as it helps to protect the civilian/military human factor, the active organological components in the area, and (b) in the environmental and the educational fields, through the formation of highly educated and specialized human resources in the academic field, able to cope with the diversity of the present operations and challenges.

5 Conclusion

The tracked robot designed for humanitarian operations represents, according to the invention, a technological product belonging to the category of tracked mobile robots, capable of replacing the human element in high-risk areas for its health and life, either by avoiding accidental detonation, or by detecting and demining anti-personnel and armor-piercing minefields in countries where there were military conflicts. The robot is able to move autonomously using electric motors which take their energy by means of electric engines that take energy from solar photovoltaic cells encased in solar panels, which is provided with a storage compartment for the explosive necessary for humanitarian demining; it has a completely modularized structure, compact, easy to mount and maintain.

Acknowledgment. The publication and the presentation of this scientific work in the plenary of the conference will be supported by UEFISCDI Romania.

References

1. Petrişor, S.M., Bârsan, Gh., Simion, Mihaela, Virca, I., Moşteanu, D.E.: Tracked robot for humanitarian demining operations. Official Bulletin of Industrial Property, Patent Section, No. 12, p. 18. OSIM, Bucharest (2017). ISSN2065-2100
2. Selig, J.M.: Geometric fundamentals of robotics – series: Monograph in Computer Science. Springer, London (2004). ISSN 978-0387208749
3. Manseur, R.: Robot modeling and kinematics. Da Vinci Engineering Press, Thomson-Delmar Learning, Paris (2006). ISBN 978-1-58450-851-9
4. Humanitarian demining. http://www.humanitarian-demining.org/2010Design/resources/NemesisM3_FS-13Feb2017.pdf. Accessed 14 Jan 2020
5. Army-technology. www.army-technology.com/projects/teodor-explosive-ordnance-eod-robot/. Accessed 14 Jan 2020
6. Galliott, J.: Military Robots – Mapping the moral landscape. Taylor & Francis Group, New York (2015). ISBN 978-1472426628

Maneuvers Under Estimation of Human Postures for Autonomous Navigation of Robot KUKA YouBot

Carlos Gordón[1](✉), Santiago Barahona[1], Myriam Cumbajín[2], and Patricio Encalada[3]

[1] Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato, UTA, 180150 Ambato, Ecuador
cd.gordon@uta.edu.ec, santiagobarahona7@gmail.com

[2] Facultad de Ingeniería y Tecnologías de la Información y la Comunicación, Universidad Tecnológica Indoamérica, UTI, 180103 Ambato, Ecuador
myriamcumbajin@uti.edu.ec

[3] Facultad de Ingeniería Mecánica, Escuela Superior Politécnica de Chimborazo, ESPOCH, 060106 Riobamba, Ecuador
patricio.encalada@espoch.edu.ec

Abstract. We present the successful demonstration of the Autonomous navigation based on maneuvers under certain human positions for an omnidirectional KUKA YouBot robot. The integration of human posture detection and navigation capabilities in the robot was successfully accomplished thanks to the integration of the Robotic Operating System (ROS) and working environments of open source library of computer vision (OpenCV). The robotic operating system allows the implementation of algorithms on real time and simulated platforms, the open source library of computer vision allows the recognition of human posture signals through the use of the Faster R-CNN (regions with convolutional neural networks) deep learning approach, which for its application in OpenCV is translated to SURF (speeded up robust features), which is one of the most used algorithms for extracting points of interest in image recognition. The main contribution of this work is that the Estimation of Human Postures is a promise method in order to provide intelligence in Autonomous Navigation of Robot KUKA YouBot due to the fact that the Robot learn from the human postures and it is capable of perform a desired task during the execution of navigation or any other activity.

Keywords: Maneuvers · Human postures · Autonomous · Navigation · KUKA YouBot

1 Introduction

Robotics is the field of science that is responsible for studying the construction, programming and design of robots [1] that when interacting with engineering fields such as: electronics, mechanics, control systems, artificial intelligence, neural networks, can

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 335–345, 2021.
https://doi.org/10.1007/978-3-030-55180-3_25

generate systems of specific purposes, to work in different areas thanks to the autonomous navigation [2].

Wide range of approaches have been developed based on autonomous navigation of robots in order to provide service for human benefits. Traffic signals recognition with Path Planning is one of the basements for autonomous navigation [3], in which the integration of Robot Operating System, MATrix LABoratory software and Open Source Computer Vision Library is the main approach. Similarly, Human rescue service is provided thanks to the autonomous navigation by using the integration of open source and Single Shot Detector algorithm [4]. Also, advanced techniques like Objects Detection and Deep Learning Networks have been used with the intention of providing the challenging autonomous navigation approach for robots [5].

With this premise, to implement a navigation with the autonomous robot KUKA YouBot [6], it is prudent to know the physical characteristics, the software that it handles and the different communication interfaces that the robot has, these are obtained by analyzing manuals of the KUKA platform itself and background of works related to the subject in repositories of national and international universities [7].

Most of the applications made with KUKA YouBot, are from previously programmed navigations on your computer where instructions from route planning are loaded from the YouBot API, sending data such as position or angular speed to the platform and KUKA arm [8]. The application previously programmed is obviously limited to have a navigation that adjusts to the needs of the operator in case of needing an extra route. But, for this work the application is improved by performing an autonomous navigation controlled by an artificial vision system that oversees sending route orders and being able to perform the maneuvers taken from human postures [9] that are needed to complete a task.

The main objective of this research is to provide Maneuvers Under Estimation of Human Postures for Autonomous Navigation of Robot KUKA YouBot, with this approach the Robot, not only is able to detect and avoid the object, but also it provides more intelligence to the robot due to the fact that it is able to identify the posture and perform a smart navigation. In order to comply this work, the integration of the Robotic Operating System (ROS) [10] and working environments of open source library of artificial vision (OpenCV) [11] with SURF (speeded up robust features) algorithm [12] were required. So that the artificial vision system can detect the signals generated by human postures and send this data to the KUKA YouBot for providing smart autonomous navigation.

2 Methodology

The present research project is emphasized in carrying out autonomous navigation of the KUKA YouBot Robot, using commands generated by an artificial intelligence system suitable to control and plan efficiently the trajectories of an autonomous omnidirectional robot, for the improvement of free movement in a Difficult access environment. The artificial vision system that runs inside the NVIDIA Jetson Nano Developer Kit is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing [13]. The Developer Kit allows to obtain the human positions of the robot operator and send

the data wirelessly to the autonomous robot KUKA YouBot, so that the human operator can say the route he wants that the robot takes to fulfill the desired task. The methodology used in this project is detailed below:

2.1 Global Analysis of Artificial Vision

The central point of the artificial vision is to be able to replace human vision, understanding. It is a discipline with which we can understand, analyze and capture objects or images from a specific place with the help of a camera and a central computer.

A. Object Tracking and Detection. Object tracking refers to the process of following a specific object of interest, or several objects, in each scene. Traditionally, it has video applications and real-world interactions where observations are made after an initial detection of objects. Now, for example, it is crucial for autonomous driving systems, such as the self-driving vehicles of companies like Uber and Tesla [14].

B. Object Detection Algorithms. The functionality of the algorithms is to detect or track objects in a short time with optimal image processing, object detection applications cover multiple and diverse industries, from permanent surveillance to real-time vehicle detection in smart cities [15]. In Table 1, all object detection algorithms with their main characteristics are sketched.

Table 1. Comparison of object detection algorithms.

Algorithm	Characteristics	Prediction time/Image	Limitations
CNN Convolutional Neural Network	Divide the image into several regions Then classify each region into several classes	More than 1 min	You need a lot of regions to predict accurately High calculation time
R-CNN Region with Convolutional Neural Network	Use selective search to generate regions. Extract about 2000 regions from each image	40–50 s	High calculation time, also uses three different models to make predictions
Fast R-CNN	Each image is passed only once to CNN and feature maps are extracted. Combine the three models used in RCNN together	2 s	Selective search is slow and therefore the calculation time is still high
Faster R-CNN	It replaces the selective search method with the region proposal network that made the algorithm much faster	0.2 s	Calculation time and, since there are different systems that work one after another

For the detection of human postures, it has been decided to implement the algorithm of "Convolutional neural network based on the fastest region" (Faster R-CNN) which mainly looks for reducing the running time in detection [16]. We select this approached because we will work on a set of data related to detection and tracking of objects, this is where Deep learning models play a vital role as it can classify and detect the different postures that the person makes [17].

The solution presented begins with the capture of the body image by means of a webcam considering the lighting and the background which are very important for the result. An intuitive acceleration solution is to integrate the regional proposal algorithm into the CNN model. R-CNN FASTER is doing exactly this: it builds a unique and unified model composed of RPN (region proposal network) and Faster R-CNN with layers of shared convolutional features [18].

2.2 Analysis of the Autonomous Robot KUKA YouBot

KUKA YouBot has the characteristic of having maximum maneuverability on a flat surface, it can maneuver in any direction without the need to reorient, this is a great advantage compared to another robots. Omnidirectional robots can be assembled with three, four or more omnidirectional wheels. These robots have four omnidirectional wheels and this leads to them having complex mechanics and control with the great advantage of greater stability and traction. The KUKA YouBot mobile platform is shown in Fig. 1.

2.3 Development and Implementation of the Selected Algorithm

In Fig. 2, the architecture of the complete system can be seen, the artificial vision system mounted on the NVIDIA Jetson Nano Developer Kit is shown, working with the ROS-MELODIC because the Jetson nano system works with the Linux-UBUNTU distribution 18.04, the camera that will collect the person's data is processed through OpenCV, the communication with the autonomous KUKA robot is through the 802.11 communication protocol applying the ROS nodes, taking ROS – HYDRO installed in the autonomous robot as the master node.

OpenCV and ROS Integration. To be able to work with ROS obtaining the OpenCV facilities, you need to install the Vision_OpenCV stack, which provides a package from the popular OpenCV library for ROS. The integration of OpenCV and ROS is depicted in Fig. 3.

- OpenCV IpIImage: Vision_OpenCV provides several packages.
- CvBridge: Bridge between ROS and OpenCV messages.
- ROS Image Message: collection of methods to treat image and pixel geometry.

Fig. 1. KUKA YouBot mobile platform.

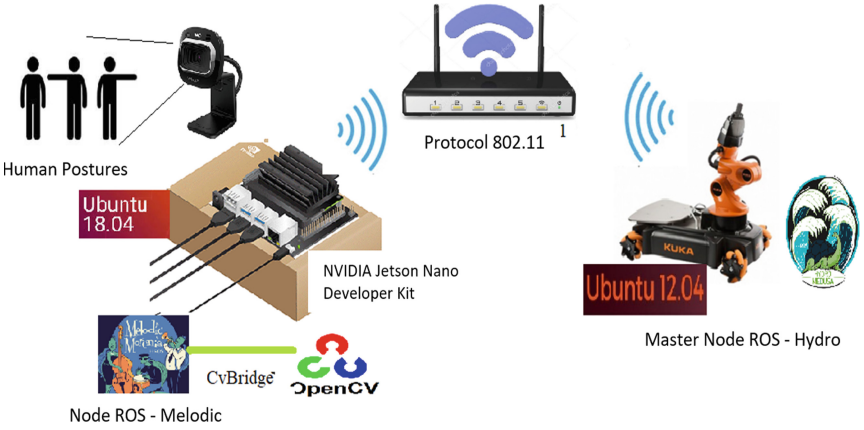


Fig. 2. Architecture of the complete system.

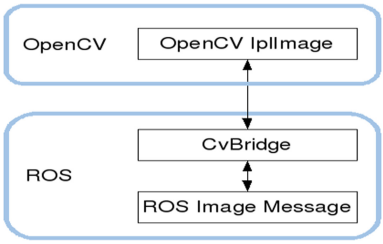


Fig. 3. Integration of OpenCV and ROS.

3 Results

Once having all the previous components, such as the interaction between OpenCV-ROS-KUKA YouBot, the system takes the data from the camera managed in OpenCV and processes it by sending data through Cv_Bridge which is the interaction bridge with ROS this creates a node Subscriber loaded in the NVIDIA Jetson Nano Developer Kit and this connects to KUKA YouBot that creates the Publisher swim and sends the speed and position data to the KUKA API. The Interaction of the complete integration is shown in Fig. 4.

ROS Node: Image processing converts ROS images to OpenCV format or vice versa through CvBridge, a library that allows you to send or receive images with OpenCV image processing. In addition, this node obtains images with the subscribers of the editors established in the ROS Nod: User application and sends different commands with its editor to the subscriber in the ROS node: controller_node.

User Application: Executes communication between the Client and the Server through the ROS Action Protocol, which is built on ROS messages. Then, the client and the server provide a simple API (application program interface, which is a set of routines, protocols and tools to create software applications) for users to request objectives (client-side) or execute objectives (server side) through function calls and callbacks. User_application and controller nodes communication provide the controller node with logical commands to be interpreted as physical actions. ROS Action Clients send the position and trajectory information processed with the API and other tools and protocols to the Action Server of the controller node. Meanwhile, the ROS editor of the User_application node sends the commands as speed.

Controller Node: Transforms commands into measurements or signals that can be understood by robot actuators.

YouBot Hardware: It is the space where the robot system is represented as a combination of decoupled functional subsystems. The manipulator arm and the base platform are arranged as the combination of several joints. At the same time, each joint is defined as a combination of an engine and a gearbox. Communication with the hardware and the controller is done through the Serial Ethercat connection.

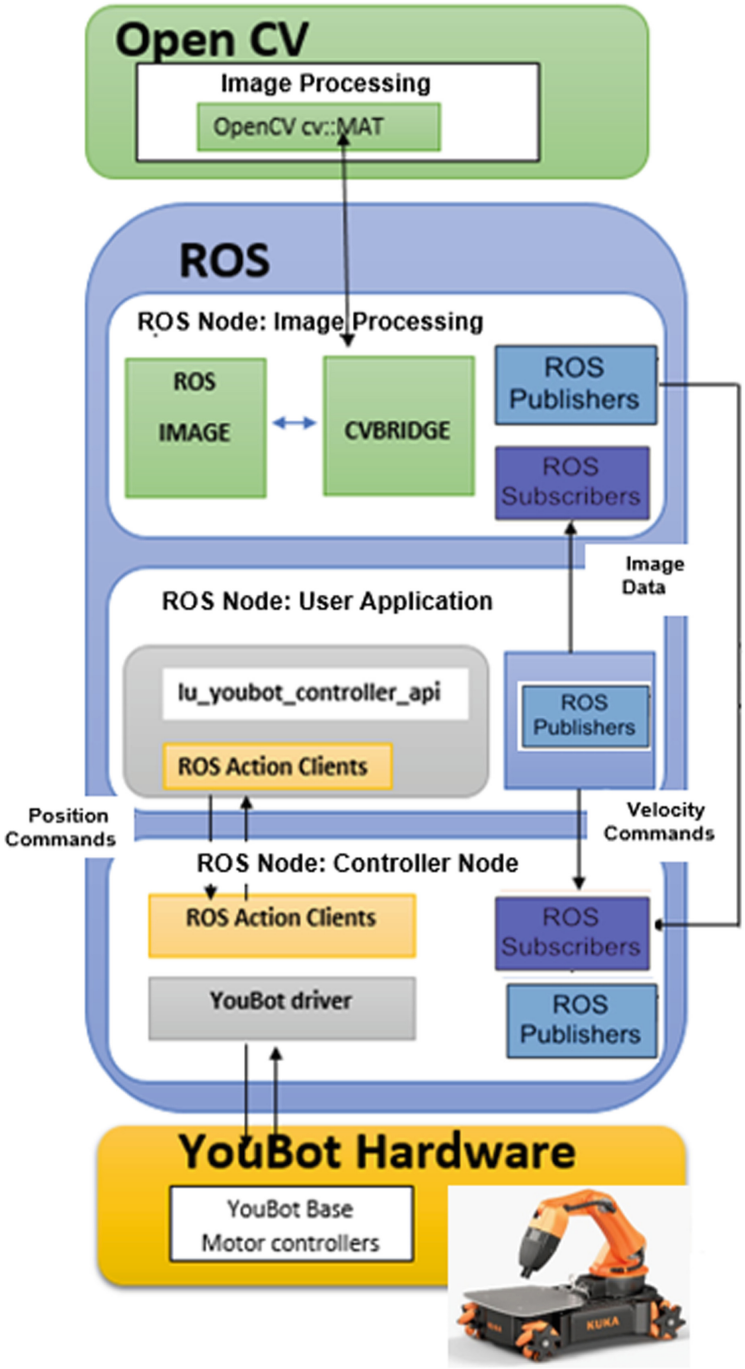
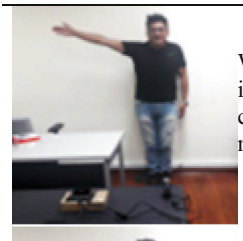
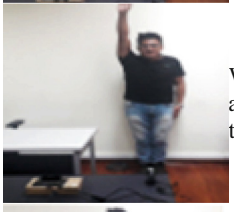
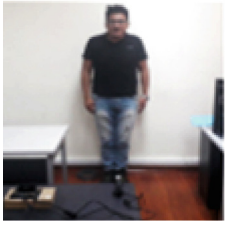


Fig. 4. Interaction of the complete integration.

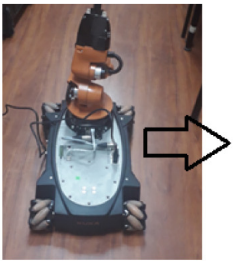
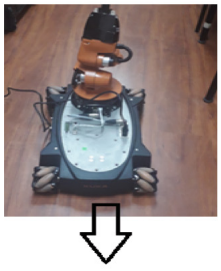
For the recognition of human postures, OpenCV creates a space of 2D descriptors, where the angles of the shoulder and the elbow are obtained, choosing the positions by means of lines of code which one wishes to write. These values are from the body located in different ways using the arms located laterally and only making movements that do not detach from the image plane. The total analysis of the human posture is detailed in Table 2.

Table 2. Human posture analysis.

Posture	Action
	With this posture Open CV takes the inclination of the right arm with the legs closed and interprets that the robot must move to the left.
	With this posture Open CV takes the inclination of the left arm with the legs closed and interprets that the robot must move to the right.
	With this posture Open CV takes the right arm up with the legs closed and interprets that the robot must move forward.
	With this posture Open CV takes both arms down with the legs closed and interprets that the robot must move backward.

The total analysis of the human posture with the loaded image, the points of interest and the actions of the KUKA YouBot robot are detailed in Table 3. In which, the action of the movement of the robot is also described, in which we can identify the main movements like move to the left, move to the right, move forward, and move backward.

Table 3. Human posture and actions of Robot KUKA YouBot.

Posture	Points of interest	Action of KUKA YouBot
		 The inclination of the right arm with the legs closed. The robot must move to the left
		 The inclination of the left arm with the legs closed. The robot must move to the right.
		 The right arm up with the legs closed. The robot must move forward.
		 Both arms down with the legs closed. The robot must move backward.

4 Conclusions

We have presented the successful demonstration of the Autonomous navigation based on maneuvers under certain human positions for an omnidirectional KUKA YouBot robot. The integration of human posture detection and navigation capabilities in the robot was successfully accomplished thanks to the integration of the Robotic Operating System (ROS) and working environments of open source library of computer vision (OpenCV). The Faster R-CNN algorithm uses a selective search that generates regions of interest, taking characteristic maps of an image generating a set of proposals of objects, within this algorithm we have several procedures in the form of a cascade that result in an optimal detection and the most important a response time approximately 0.2 s. As a result, the estimation of Human Postures is a promise method in order to provide intelligence in Autonomous Navigation of Robot KUKA YouBot due to the fact that the Robot learn from the human postures and it is capable of perform a desired task during the execution of navigation or any other activity.

Acknowledgments. The authors thank the Technical University of Ambato and the "Dirección de Investigación y Desarrollo" (DIDE) for their support in carrying out this research, in the execution of the project "Plataforma Móvil Omnidireccional KUKA dotada de Inteligencia Artificial utilizando estrategias de Machine Learnig para Navegación Segura en Espacios no Controlados", project code: PFISEI27.

References

1. Dailami, F., Melhuish, C., Cecchi, F., Leroux, C.: Robotics innovation facilities. In: Advances in Robotics Research: From Lab to Market, pp. 29–45. Springer, Cham (2020)
2. Gordón, C., Encalada, P., Lema, H., León, D., Castro, C., Chicaiza, D.: Intelligent autonomous navigation of robot KUKA YouBot. In: Proceedings of SAI Intelligent Systems Conference, pp. 954–967. Springer, Cham (2019)
3. Gordón, C., Encalada, P., Lema, H., León, D., Peñaherrera, C.: Autonomous robot KUKA YouBot navigation based on path planning and traffic signals recognition. In: Proceedings of the Future Technologies Conference, pp. 63–78. Springer, Cham (2018)
4. Gordón, C., Lema, H., León, D., Encalada, P.: Human rescue based on autonomous robot kuka youbot with deep learning approach. In: 2019 Sixth International Conference on eDemocracy and eGovernment (ICEDEG), pp. 318–323. IEEE (2019)
5. Gordón, C., Encalada, P., Lema, H., León, D., Castro, C., Chicaiza, D.: Autonomous robot navigation with signaling based on objects detection techniques and deep learning networks. In: Proceedings of SAI Intelligent Systems Conference, pp. 940–953. Springer, Cham (2019)
6. KUKA YouBot Homepage. http://www.youbot-store.com/developers/software. Accessed 08 Jan 2020
7. Shardyko, I., Dalyaev, I., Nanyageev, I., Shmakov, O.: Inverse kinematics solution for robots with simplified tree structure and 5-DoF robot arms lacking wrist yaw joint. In: Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings", pp. 113–124. Springer, Singapore (2020)
8. Ahmed, S., Popov, V., Topalov, A., Shakev, N.: Hand gesture based concept of human-mobile robot interaction with leap motion sensor. IFAC-PapersOnLine **52**(25), 321–326 (2019)

9. Takano, W., Haeyeon, L.E.E.: Action description from 2D human postures in care facilities. IEEE Robot. Autom. Lett. **5**(2), 774–781 (2020)
10. Robot Operating System (ROS) Homepage. https://www.ros.org/. Accessed 07 Jan 2020
11. Open source computer vision (OpenCV) Homepage. https://opencv.org/. Accessed 06 Jan 2020
12. Bay H., Tuytelaars T., Van Gool L.: SURF: speeded up robust features. In: Leonardis A., Bischof H., Pinz A. (eds) Computer Vision – ECCV 2006. ECCV 2006. LNCS, vol. 3951. Springer, Heidelberg (2006)
13. NVIDIA Homepage. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/. Accessed 07 Jan 2020
14. TESLA Homepage. https://www.tesla.com/. Accessed 08 Jan 2020
15. Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., Ma, X.: 3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. Sustain. Cities Soc. **54**, 102002 (2020)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015)
17. Tucker, M., Aksaray, D., Paul, R., Stein, G.J., Roy, N.: Learning unknown groundings for natural language interaction with mobile robots. In: Robotics Research, pp. 317–333. Springer, Cham (2020)
18. Zhu, X., Chen, C., Zheng, B., Yang, X., Gan, H., Zheng, C., Yang, A., Mao, L., Xue, Y.: Automatic recognition of lactating sow postures by refined two-stream RGB-D faster R-CNN. Biosys. Eng. **189**, 116–132 (2020)

Towards Online-Prediction of Quality Features in Laser Fusion Cutting Using Neural Networks

Ulrich Halm[1(✉)], Dennis Arntz-Schroeder[2], Arnold Gillner[2,3], and Wolfgang Schulz[1,2]

[1] Nonlinear Dynamics of Laser Processing, RWTH Aachen University, Aachen, Germany
ulrich.halm@nld.rwth-aachen.de
[2] Fraunhofer Institute for Laser Technology, Aachen, Germany
[3] Chair for Laser Technology, RWTH Aachen University, Aachen, Germany

Abstract. The fine-scaled striation structure as a relevant quality feature in laser fusion cutting of sheet metals cannot be predicted from online process signals, today. High-speed recordings are used to extract a fast melt-wave signal as temporally resolved input signal and a surrogate surface profile as output. The two signals are aligned with a sliding-window algorithm and prepared for a one-step ahead prediction with neural networks. As network architecture a convolutional neural network approach is chosen and qualitatively checked for its suitability to predict the general striation structure. Test and inference of the trained model reproduce the peak count of the surface signal and prove the general applicability of the proposed method. Future research should focus on enhancements of the neural network design and on transfer of this methodology to other signal sources, that are easier accessible during laser cutting of sheet metals.

Keywords: Time series forecasting · Convolutional neural networks · Laser fusion cutting · Signal processing

1 Introduction

Machine learning approaches to enhance productivity and quality in industrial laser cutting are not very common today. Applications usually focus on classification of parameter sets [1] or in-process detection of faulty parts [2]. Besides online detection of faulty parts, i.e. unsuccessful cuts, there is a strong interest in online prediction of relevant quality features of the cut. Pronounced striation patterns as shown in Fig. 1 are caused by irregular melt expulsion during laser fusion cutting of sheet metals and result in mean surface roughnesses R_{z5} of several 10 µm. Although sheet metal cutting is one of the most important applications of industrial lasers (total revenues for metal cutting systems equal B$1.5 out of B$4.3 total for industrial lasers in 2017 [3]), the exact mechanism

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 346–359, 2021.
https://doi.org/10.1007/978-3-030-55180-3_26

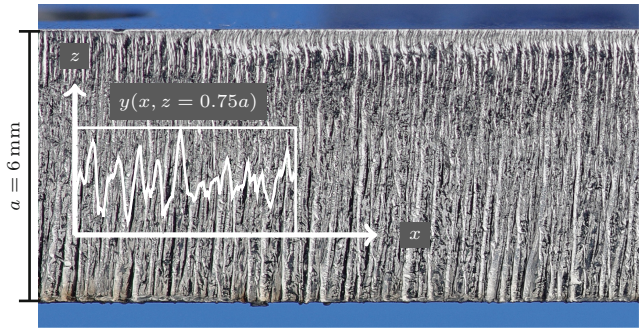


Fig. 1. Cut surface in laser cutting of stainless steel with thickness $a = 6\,\mathrm{mm}$. The surface profile $y(x, z)$ shows a certain roughness due to instabilities in the melt film during cutting.

of striation formation is not fully understood yet. There are various theoretical [4,5] and diagnostic [6,7] studies to extend the process understanding and derive measures to reduce the mean surface roughness.

The work presented in this paper aims on the interconnection between melt wave dynamics and surface roughness near the top of the cut surface by means of machine learning. For this purpose high speed recordings of laser cuts, which incorporate the so-called trim-cut technique [7] (Fig. 2) to provide optical access to the process, are used to train convolutional neural networks. To predict the surface structure, two signals are extracted from the high speed recordings:

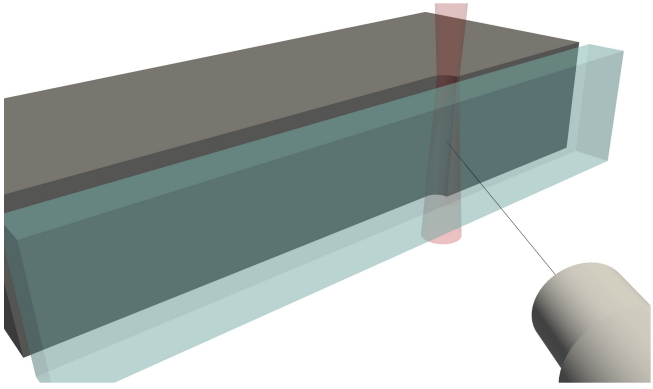


Fig. 2. Sketch of trim cut technique: the cut is performed along the edge of a metal sheet. The missing half of the material is replaced by a transparent substrate to preserve the major behavior of the process. This enables high speed video recordings of the melt film dynamics.

(1) Melt-wave signal: an integrated brightness signal that represents the melt film dynamics in the upper 20% of the cut (INPUT).
(2) Surface signal: a brightness signal of the cut surface, that represents the basic structure of the striations at a certain height of the cut (OUTPUT).

First studies presented in this paper show that a causal relation between the history of a melt wave signal and the surface signal might be found by convolutional neural networks. Current research aims on the transfer of this procedure to signal sources that are accessible during real laser cuts, like coaxial process observation optics or photo-diodes. By analyzing which features of the signal are considered as important by the neural network or by analyzing the effect of additional noise at certain parts of the input signal, the process understanding will be extended in future research.

The purpose of this paper is to present and prove the general applicability of a novel method to analyze sensor data in laser fusion cutting on a sub-millisecond time scale using neural networks to predict the general striation structure. Upcoming work aims at

- deepening the understanding of the correlation between melt wave dynamics and striations by analyzing the features extracted by the neural network,
- derive measures for closed-loop control in laser fusion cutting to reduce striations.

2 Related Work

Time series forecasting using neural networks usually works on a one-step-ahead mechanism, so that input data from sensors of a certain duration until the current time step is used as input. The output is one scalar value per output quantity for the next time step. Both, input and output, can consist of multiple data sources, also called features.

Horelu et al. [8] focused on recurrent neural networks (RNN) with long-short term memory (LSTM) architecture for time series with long term dependencies and achieved high accuracy in one- and multi-step ahead prediction of temperature values. As RNNs have an integrated memory ability and thus seem to be the first choice for time series forecasting, simpler approaches, like multi-layer perceptrons (MLP) and convolutional neural networks (CNN) offer clear advantages over recurrent approaches in terms of robustness and training speed.

Binkowski et al. [9] presented a method to predict time series data from asynchronous input data using *Significance-Offset CNNs*. This method is based on CNNs and is designed for multi-variant asynchronous inputs in the presence of noise. Binkowski et al. also compared the performance of standard CNN and LSTM approaches for synchronous input data and both result in nearly the same prediction error (Mean squared error 0.029 vs. 0.028 for synchronous data set).

Koprinska et al. [10] compared the errors of different NN designs while predicting solar and electricity data for particular countries. They found clear advantages of MLP and CNN designs over RNN approaches. The selected LSTM

method resulted in a greater error, than a plain persistence model, that just uses the last time step as prediction for the next one. Besides bad prediction quality in this example, a single LSTM model cannot be calculated in parallel and thus cannot benefit from modern multi-processor architectures. As a consequence training durations for LSTMs are usually several orders of magnitude greater than for CNN or MLP networks.

Kim and Cho [11] predict residential energy consumption in Korea using combined CNN-LSTM neural networks. They use a sliding window algorithm [12] to forecast power consumption from multivariate input data with a combination of convolutional and pooling layers to filter noise and LSTM layers followed by dense layers to model the irregular temporal behavior. The proposed linear combination of CNN and LSTM provided better prediction accuracy than standard LSTM approaches.

Closed-loop control in laser fusion cutting using coaxial process observation is presented by Peng et al. [13]. Current approaches focus on retaining process parameters inside stable domains.

For the planned online prediction method for surface profiles in laser cutting we collect:

- One-step ahead forecasting is performed with a sliding window algorithm.
- CNNs are a valid starting point for time series analysis and have proven to be more robust and to be trained faster than LSTMs.
- Future approaches should combine CNN and LSTM layers.

3 Methodology

The proposed method uses 1-dimensional input data to predict the structure of the cut surface with a one-step ahead sliding window technique. To train a neural network, we need to extract temporal input data from a signal source and align it to the output data, which is the surface profile in this work. As starting point and to proof the general applicability of the concept, we use high speed recordings from trim-cut diagnosis. Future applications should utilize sensor signals, that can be retrieved from the real process, i.e. signals from photo-diodes or coaxial process monitoring systems.

3.1 Signal Preparation

Figure 3 shows a single frame of a trim-cut recording. The sheet thickness is $a = 6\,\text{mm}$, the recording frame rate is $f_r = 70\,\text{kHz}$, the image size is $132 \times 298\,\text{px}^2$. The resulting length scale is $l_0 = 43.5\,\text{px/mm}$ and the cutting speed translates to $v_{surf} = 59.04\,\text{frames/px}$ in horizontal direction, which means that every 59.04 video frames the work piece should have moved 1 pixel to the left. The main part of the trim-cut recording (red region in Fig. 3 b) shows the slow moving cut surface as gray-scale video-signal. The front part (green region) shows the fast motion of the melt waves flowing downwards. Figure 4 shows 6 video frames with an offset of 20 video frames each.

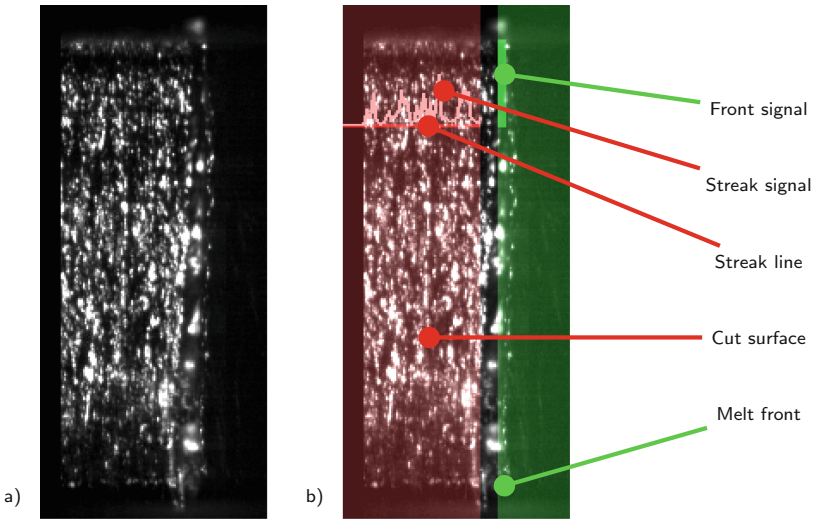


Fig. 3. Extraction of front and surface signal from trim cut recordings. a) Plain video frame. b) Highlighted features.

Extraction of Surface Signal (Output). The high speed recording in Fig. 3a) shows a gray-scale image of the cut surface and thus the intensity of the reflected light. It depends on the direction of the surface normal, the viewing direction and the source of light. As the viewing direction is constant and we assume an ambient light source, the distribution of intensity reflects the surface structure. The height of the surface profile cannot be reconstructed from this signal, but basic information like striation wavelength and change of striation wavelength with depth of cut can be derived. We select a fixed depth of cut $z_p = 0.2a$ to extract the surface profile $y(x, z = z_p)$ along a horizontal line. The gray-scale signal inside the slow moving part of the image is extracted at this depth (red line in Fig. 3b) and interpreted as surface signal for this time step (light-red profile). By aligning these signals from all video frames in time, a surrogate surface profile by a temporal median value can be extracted. Figure 5 shows this surface profile for a recording of $N = 15\,500$ frames. The total cutting distance L equals

$$L = v_0 \frac{N}{f_r} = 6.6\,\mathrm{mm} \tag{1}$$

with a cutting speed of $v_0 = 1.8\,\mathrm{m\,min^{-1}}$. The comparatively high peaks at the end of the signal for frames $n \geq 14\,000$ are due to the sudden end of the recording and the fact, that we use a temporal median gray-scale value. This overestimates the peak height and such the rear part of the surface signal is not usable for training of the neural network. However, we might use it for inference to check, if the general structure of the signal (wave length) is reproduced.

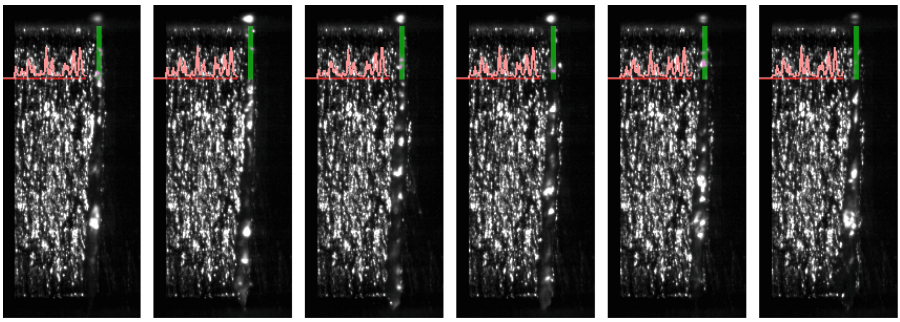


Fig. 4. Extraction of front and surface signal from trim cut recordings. a) Plain video frame. b) Highlighted features.

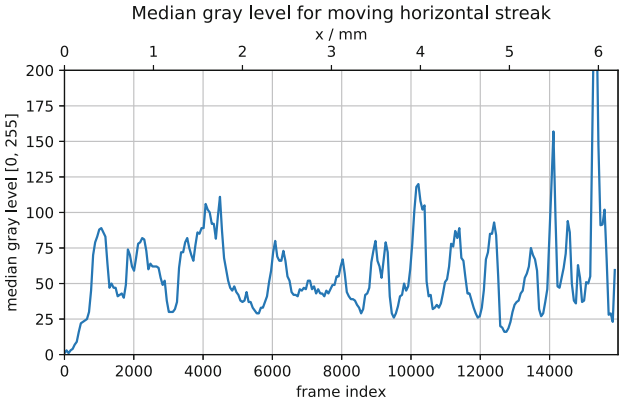


Fig. 5. Extracted surrogate surface signal, represented by temporal median gray-scale value.

This is possible because the input signal does not suffer from this overestimation as it is retrieved differently.

Because the surface signal is moving comparatively slow with a dimensionless rate of $v_s \approx 1/60$ pixels per frame, the total amount of discrete surface profile data points is $N_s = 350$. This value is greater than $N/v_s = 258$ because of the initial signal length that is observable in the recording.

Extraction of Front Signal (Input). The selection of the INPUT-signal source for the neural network is motivated by the physical task, that is be analyzed here: How does the dynamics of the melt waves at the top of the cutting front affect the structure of the cut surface? For this purpose we extract an integrated signal as first guess for the melt film dynamics. As the melt waves are running downwards, we select only melt wave data that lies above the extracted profile $z \geq z_p$. The dark green area (*Front signal*) in Fig. 3 indicates this integration region.

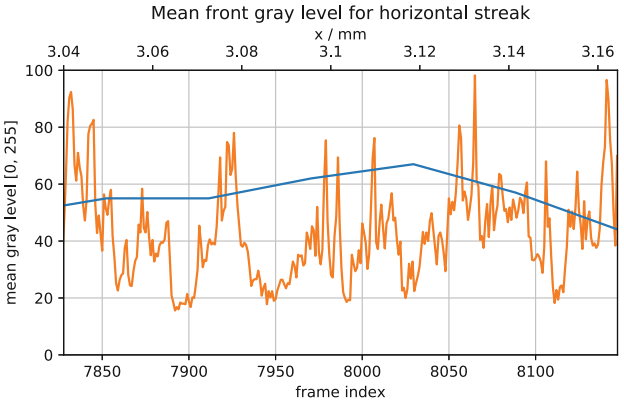


Fig. 6. Extracted front signal for frames $0.49N \leq n \leq 0.51N$. The orange curve shows the evolution of the fast melt wave signal. The blue curve shows the slow evolution of the aligned surrogate surface signal.

The motivation to use an integrated signal is driven by the perspective to use real process signals in future works, like photo-diode signals, which can be recorded in practice, easily. As the norm of the integrated signal is the size of the integration area, the front signal is retrieved as mean value inside this area. Figure 6 shows the dynamics of the front signal for the central 2% of the input frames. The aligned surface signal is plotted in blue and the wavelength of the OUTPUT-Signal can be recognized as at least one order of magnitude greater than the INPUT-Signal.

3.2 Preparation of Data

We use a sliding window approach of the melt front signal (INPUT X_i^n) to perform a one-step ahead prediction of the surface profile (OUTPUT Y^n), with $i \in [0, N_W - 1]$ the frame index inside the time series window and time series index $n \in [0, N_S - 1]$. The window size N_W is estimated as at least one full period of the (OUTPUT) signal, for the analyzed recording, we set $N_W = 2000$. Due to the slow moving surface profile, the amount of discrete output data points is $N_S = 350$. Because of the moving window algorithm with $N_W > 0$, the first $N_W/v_s = 33$ data points of the output signal cannot be used for training.

Each output data point Y^n is aligned with an input vector $X_i^n, i \in [0, N_W - 1]$, so that

$$t(Y^n) \geq t(X^n_{N_W - 1}), \tag{2}$$

where $t(\cdot)$ is the time for a specific data point, represented by the frame index.

3.3 Model Setup

We implement the NN in python using the Keras [14] open-source library as front-end for tensorflow-GPU. As selected NN architecture we start with convolutional neural networks. CNNs are well suited for pattern recognition and

Table 1. Proposed CNN model architecture

Layer Type	# Filter	Kernel size	Stride	Rate	# Parameters
Conv1D	200	10	1		2200
ReLU	–	–	–	–	–
Dropout	–	–	–	0.2	–
Conv1D	100	10	1	–	200100
ReLU	–	–	–	–	–
Dropout	–	–	–	0.2	–
MaxPooling1D	–	30	2	–	–
Conv1D	100	20	1	–	200100
ReLU	–	–	–	–	–
Dropout	–	–	–	0.2	–
Conv1D	200	10	1	–	200200
ReLU	–	–	–	–	–
Dropout	–	–	–	0.2	–
Flatten	–	–	–	–	–
Dense	1	–	–	–	189801
Total parameters					792401

have been applied to time series forecasting successfully. CNNs are also easier to implement and exhibit a more robust convergence behavior when the hyper-parameters of the model are changed. Table 1 shows the selected model architecture. It consists of several connected convolutional layers that extract features from an input vector. If x_{ij}^{nk} is the input vector for data sample n and layer k, then the output y_{ij}^{nk} of the j^{th} feature map of a convolutional layer is given by

$$y_{ij}^{nk} = \sigma^k \left(b_j^k + \sum_{m=0}^{M^k - 1} w_{mj}^k x_{i+m,j}^{nk} \right), \tag{3}$$

where m is the index value of the filter. The function $\sigma^k(\cdot)$ is the activation function of layer k and is responsible for the non-linear behavior of the neural network. In the proposed work, we select

$$\sigma^k(\cdot) = ReLU(\cdot) = \max(0, \cdot) \tag{4}$$

for all layers, except the final collection layer. The pooling layer is a typical feature of CNNs and reduces the outputs of a cluster of neurons to a single pooling value, in the case of *MaxPooling* to the maximum value. This drastically reduces the number of parameters for the next layer and lowers the risk of

overfitting. In image recognition, pooling layers enhance the robustness of the model against translation of individual features. The pooling output of layer k for feature map j using output of layer $k-1$ is

$$p_{ij}^{nk} = \max\left(y_{i\cdot s+m,j}^{n,k-1},\ m \in [0, M^k - 1]\right) \tag{5}$$

with stride multiplier $s \in \mathbb{N}_{>0}$ and pooling size M^k of layer k.

The *Dropout* layers added after each convolution layer l with dropout rate r^l randomly set the values of a fixed fraction of output neurons to zero. This technique is used to prevent overfitting.

The final layer (*Dense*) sums up the flattened outputs of all feature maps j with weights w_{ij}^k and bias b^k. Because we do regression – output is one scalar value for one-step ahead prediction – and not classification with this model, no activation function is applied to the final layer.

4 Findings

Fig. 7. Convergence of mean square error during training of neural network.

The input data time series X_i^n and output scalar values Y^n with $n < 161$ are fed into the fitting routine of the model. Figure 7 shows the convergence of the mean squared error

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}\left(Y^n - y^{n,L-1}\right)^2 \tag{6}$$

for training and validation data, where $y^{n,L-1}$ is the prediction for data set n as output of the final layer $L-1$. For validation the final 20% of the input data set is used ($n > 128$). The batch size has been set to the amount of data sets, the

number of training epochs to 1000. *Adam* [15] was selected as optimizer with a learning rate of $\alpha = 0.001$. All X^n and Y^n were scaled to the range $[0, 1]$ using the global minimum and maximum values. The training was performed on a *Intel Xeon Platinum 8168* system, equipped with one *Nvidia GP104GL (Quadro P5000)* GPU and took $t_{train} = 195\,\text{s}$ for the presented model and data set.

4.1 Prediction

Figure 8 shows the unscaled prediction of the trained model (red curve) for the surrogate surface profile, the blue curve represents the objective output of the model. The vertical dashed line displays the split between training and testing data. Within the training region (left of the split line), the red curve follows the general structure of the surface profile without matching the peaks heights exactly. Within the testing region peak positions and heights are not matched correctly. However, taking the simple CNN model into account and the lack of large amounts of input data ($N = 161$), the general structure, i.e. wave-length of the signal, is met astonishingly well. Slight modifications of the model hyper-parameters given in Table 1 have no strong impact on this fundamental result of the model. This also proofs the robustness of the CNN approach. Until today the authors were not able to reproduce this result with a plain LSTM approach. A combination of CNN and LSTM is planned.

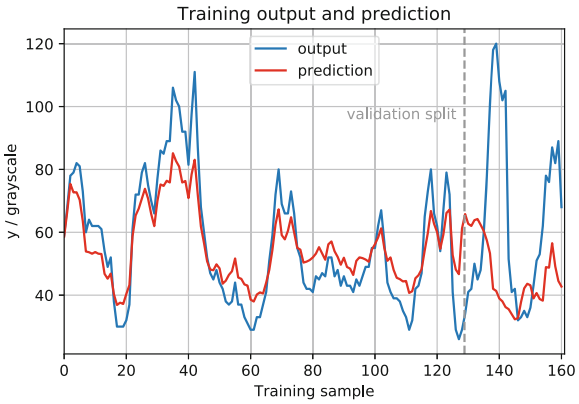


Fig. 8. Surface data and prediction of trained neural network.

4.2 Inference

Because of the extraction method of the output signal, the final data sets $n > 161$ cannot be used for training. The sudden break of the high speed recording and

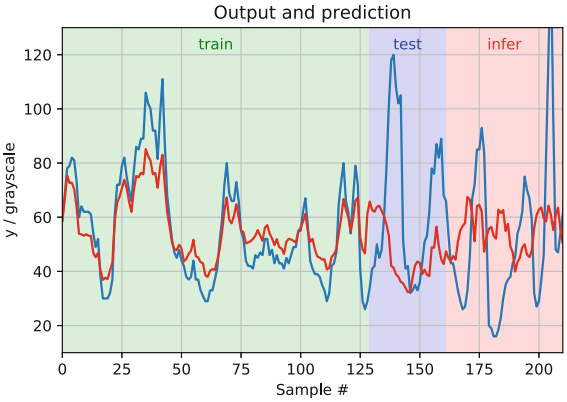


Fig. 9. Surface signal and prediction.

the reconstruction of the surface structure as temporal median values lead to overestimated peak heights. However, the general structure of the signal is valid and might be used for model inference. Figure 9 displays the inference result as red curve inside the light-red region on the right-hand side of the plot. The blue curve represents the objective output, thus the surrogate surface profile in gray-scale values. As inside the testing region, the position of the peaks is not reproduced correctly, but the count of the largest peaks matches the count of the objective signal.

This is a surprisingly good result, as we use an integrated melt wave dynamics signal from high speed recordings to predict the general surface structure with a very simple CNN approach.

4.3 Integral Vs. Local Input Signal

The INPUT time series signal has been extracted as mean value from a box-area in the high-speed recording (dark green area in Fig. 3). The motivation to integrate this area was driven by the assumption, that every melt-wave signal, that crosses this region contributes to the surface structure. Figure 10 proves, that the integrated signal provides a better qualitative description of the surface profile than a local melt-front signal. The local melt front signal is extracted from a horizontal line at the same height as the extracted surface profile. This is an important finding for upcoming studies with photo-diode signals, that provide an integrated signal inherently.

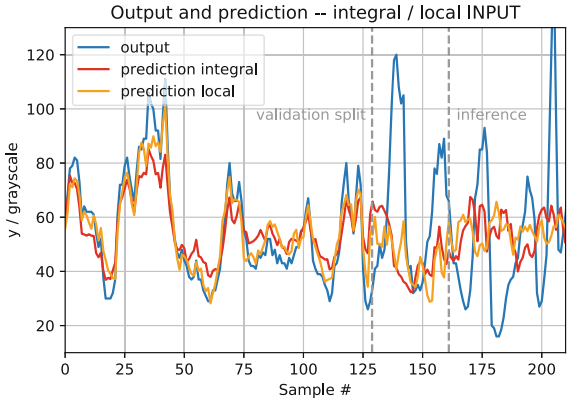


Fig. 10. Surface signal and prediction for integrated and local input data.

5 Conclusion

We present a neural network approach to predict the general structure of cut surfaces in laser fusion cutting of sheet metals. As input data, a time series representing the melt film dynamics in the upper part of the cut is extracted from trim cut recordings. The general applicability of a simple CNN approach is proven on an example data set with $N = 210$ one-step ahead forecasting time-series.

5.1 Discussion

To the author's knowledge, this is the first time that the general surface structure of laser cut surfaces can be reconstructed from online-process signals. If this method proofs its applicability on real process signals, it can drive the process understanding of laser cutting and lead to new modeling and control approaches. The one-step ahead prediction technique using convolutional neural networks has proven to be very robust and surprisingly accurate given the small size of the training data set. By analyzing which features of the input signal have been extracted by the neural network, the understanding of the melt film dynamics in laser fusion cutting will be extended in the future. A first important hint is, that an integral signal in the upper part of the cut provides at least the same prediction quality compared to a local brightness signal.

5.2 Limitations

The work presented is a prove of concept for a novel method of signal processing in laser fusion cutting based on a single trim-cut recording of melt film dynamics. The applicability for differing process parameters and other kinds of observation is expected, but has to be verified.

5.3 Future Work

Enhancements of the proposed CNN approach should incorporate the combined CNN-LSTM method presented by Kim and Cho [11]. Another option are so called fully convolutional networks (FCN) as demonstrated by Wang et al. [16] with global average pooling to identify those regions in the signal, that contribute to specific characteristics of the output. Prospective studies should use signal sources available in *real* cutting processes, like photo-diode or coaxial process monitoring signals. To deepen the understanding of the formation of striations in laser fusion cutting, the Neural Network has to be analyzed further, to understand, which characteristics of the melt film dynamics are responsible for an irregular cut surface.

Acknowledgments. All presented investigations are conducted in the context of the Collaborative Research Centre SFB1120 "Precision Melt Engineering" at RWTH Aachen University and funded by the German Research Foundation (DFG). For the sponsorship and support we wish to express our sincere gratitude.

Conflict of Interest Statement. The authors declare that there is no conflict of interest.

References

1. Tercan, H., Al Khawli, T., Eppelt, U., Büscher, C., Meisen, T., Jeschke, S.: Improving the laser cutting process design by machine learning techniques. Prod. Eng. Res. Devel. **11**(2), 195–203 (2017)
2. Santolini, G., Rota, P., Gandolfi, D., Bosetti, P.: Cut quality estimation in industrial laser cutting machines: a machine learning approach. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019
3. Belforte, D.A.: 2017 was a great year - for industrial lasers. Ind. Laser Solutions **33**(1), 11–15 (2018)
4. Kheloufi, K., Hachemi Amara, E., Benzaoui, A.: Numerical simulation of transient three-dimensional temperature and kerf formation in laser fusion cutting. J. Heat Transfer **137**(11), 112101/1–112101/9 (2015)
5. Zaitsev, A.V., Ermolaev, G.V., Polyanskiy, T.A., Gurin, A.M.: Numerical simulation of the shape of laser cut for fiber and co2 lasers. In: AIP Conference Proceedings, vol. 1893, no. 1, p. 030046 (2017)
6. Hirano, K., Fabbro, R.: Experimental investigation of hydrodynamics of melt layer during laser cutting of steel. J. Phys. D Appl. Phys. **44**(10), 105502 (2011)
7. Arntz, D., Petring, D., Jansen, U., Poprawe, R.: Advanced trim-cut technique to visualize melt flow dynamics inside laser cutting kerfs. J. Laser Appl. **29**(2), 022213 (2017)
8. Horelu, A., Leordeanu, C., Apostol, E., Huru, D., Mocanu, M., Cristea, V.: Forecasting techniques for time series from sensor data. In: 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC, pp. 261–264, September 2015
9. Binkowski, M., Marti, G., Donnat, P.: Autoregressive convolutional neural networks for asynchronous time series. CoRR, abs/1703.04122 (2017)

10. Koprinska, I., Wu, D., Wang, Z.: Convolutional neural networks for energy time series forecasting. In 2018 International Joint Conference on Neural Networks, IJCNN, pp. 1–8, July 2018
11. Kim, T.-Y., Cho, S.-B.: Predicting residential energy consumption using cnn-lstm neural networks. Energy **182**, 72–81 (2019)
12. Kim, T.-Y., Cho, S.-B.: Web traffic anomaly detection using c-lstm neural networks. Expert Syst. Appl. **106**, 66–76 (2018)
13. Wen, P., Zhang, Y., Chen, W.: Quality detection and control during laser cutting progress with coaxial visual monitoring. J. Laser Appl. **24**(3), 032006 (2012)
14. Chollet, F., et al.: Keras (2015). https://keras.io
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv e-prints, arXiv:1412.6980 (2014)
16. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 International Joint Conference on Neural Networks, IJCNN, pp. 1578–1585, May 2017

Convergence of a Relaxed Variable Splitting Method for Learning Sparse Neural Networks via ℓ_1, ℓ_0, and Transformed-ℓ_1 Penalties

Thu Dinh(✉) and Jack Xin

Department of Mathematics, University of California, Irvine, CA 92697, USA
{t.dinh,jack.xin}@uci.edu

Abstract. Sparsification of neural networks is one of the effective complexity reduction methods to improve efficiency and generalizability. We consider the problem of learning a one hidden layer convolutional neural network with ReLU activation function via gradient descent under sparsity promoting penalties. It is known that when the input data is Gaussian distributed, no-overlap networks (without penalties) in regression problems with ground truth can be learned in polynomial time at high probability. We propose a relaxed variable splitting method integrating thresholding and gradient descent to overcome the non-smoothness in the loss function. The sparsity in network weight is realized during the optimization (training) process. We prove that under ℓ_1, ℓ_0, and transformed-ℓ_1 penalties, no-overlap networks can be learned with high probability, and the iterative weights converge to a global limit which is a transformation of the true weight under a novel thresholding operation. Numerical experiments confirm theoretical findings, and compare the accuracy and sparsity trade-off among the penalties.

Keywords: Regularization · Sparsification · Non-convex optimization.

1 Introduction

Deep neural networks (DNN) have achieved state-of-the-art performance on many machine learning tasks such as speech recognition (Hinton et al. [8]), computer vision (Krizhevsky et al. [10]), and natural language processing (Dauphin et al. [3]). Training such networks is a problem of minimizing a high-dimensional non-convex and non-smooth objective function, and is often solved by simple first-order methods such as stochastic gradient descent. Nevertheless, the success of neural network training remains to be understood from a theoretical perspective. Progress has been made in simplified model problems. Shamir (2016) showed learning a simple one-layer fully connected neural network is hard for some specific input distributions [20]. Recently, several works (Tian [22]; Brutzkus and Globerson [1]) focused on the geometric properties of loss

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 360–374, 2021.
https://doi.org/10.1007/978-3-030-55180-3_27

functions, which is made possible by assuming that the input data distribution is Gaussian. They showed that stochastic gradient descent (SGD) with random or zero initialization is able to train a no-overlap neural network in polynomial time.

Another notable issue is that DNNs contain millions of parameters and lots of redundancies, potentially causing over-fitting and poor generalization [26] besides spending unnecessary computational resources. One way to reduce complexity is to sparsify the network weights using an empirical technique called pruning [11] so that the non-essential ones are zeroed out with minimal loss of performance [7,14,24]. Recently a surrogate ℓ_0 regularization approach based on a continuous relaxation of Bernoulli random variables in the distribution sense is introduced with encouraging results on small size image data sets [12]. This motivated our work here to study deterministic regularization of ℓ_0 via its Moreau envelope and related ℓ_1 penalties in a one hidden layer convolutional neural network model [1].

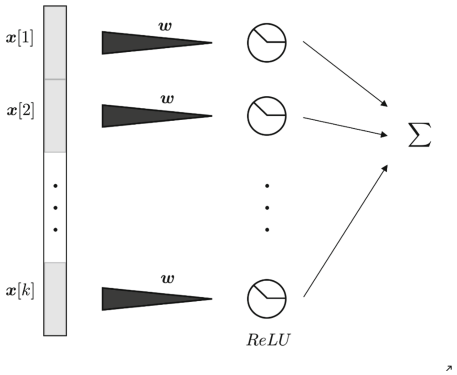


Fig. 1. The architecture of a no-overlap neural network

Our contribution: We propose a new method to sparsify DNNs called the Relaxed Variable Splitting Method (RVSM), and prove its convergence on a simple one-layer network (Fig. 1). Consider the population loss:

$$f(\boldsymbol{w}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[(L(\boldsymbol{x};\boldsymbol{w}) - L(\boldsymbol{x};\boldsymbol{w}^*))^2\right]. \tag{1}$$

where $L(\boldsymbol{x},\boldsymbol{w})$ is the output of the network with input $\boldsymbol{x}$ and weight $\boldsymbol{w}$ in the hidden layer. We assume there exists a ground truth $\boldsymbol{w}^*$. Consider sparsifying the network by minimizing the Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}) = f(\boldsymbol{w}) + \|\boldsymbol{w}\|_1 \tag{2}$$

where the ℓ_1 penalty can be changed to ℓ_0 or Transformed-ℓ_1 penalty [15,27]. Empirical experiments show that our method also works on deeper networks, since the sparsification on each layer happens independently of each other.

The rest of the paper is organized as follows. In Sect. 2, we briefly overview related mathematical results in the study of neural networks and complexity reduction. Preliminaries are in Sect. 3. In Sect. 4, we state and discuss the main results. The proofs of the main results are in Sect. 5, and numerical experiments are in Sect. 6.

2 Related Work

In recent years, significant progress has been made in the study of convergence in neural network training. From a theoretical point of view, optimizing (training) neural network is a non-convex non-smooth optimization problem, which is mainly solved by (stochastic) gradient descent. Stochastic gradient descent methods were first proposed by Robins and Monro in 1951 [18]. Rumelhart et al. introduced the popular back-propagation algorithm in 1986 [19]. Since then, many well-known SGD methods with adaptive learning rates were proposed and applied in practice, such as the Polyak momentum [16], AdaGrad [6], RMSProp [23], Adam [9], and AMSGrad [17].

The behavior of gradient descent methods in neural networks is better understood when the input has *Gaussian* distribution. In 2017, Tian showed the population gradient descent can recover the true weight vector with random initialization for one-layer one-neuron model [22]. Brutzkus & Globerson (2017) showed that a convolution filter with non-overlapping input can be learned in polynomial time [1]. Du et al. showed (stochastic) gradient descent with random initialization can learn the convolutional filter in polynomial time and the convergence rate depends on the smoothness of the input distribution and the closeness of patches [4]. Du et al. also analyzed the polynomial convergence guarantee of randomly initialized gradient descent algorithm for learning a one-hidden-layer convolutional neural network [5]. Non-SGD methods for deep learning were also studied in the recent years. Taylor et al. proposed the Alternating Direction Method of Multipliers (ADMM) to transform a fully-connected neural network into an equality-constrained problem to solve [21]. A similar algorithm to the one introduced in this paper was discussed in [13]. There are a few notable differences. First, their parameter ϱ (respectively our parameter β) is large (resp. small). Secondly, the update on $\boldsymbol{w}$ in our paper does not have the form of an argmin update, but rather a gradient descent step. Lastly, their analysis does not apply to ReLU neural networks, and the checking step will be costly and impractical for large networks. In this paper, we will show that having β small is essential in showing descent of the Lagrangian, angle, and giving a strong error bound on the limit point. We became aware of [13] lately after our work was mostly done.

3 Preliminaries

3.1 The One-Layer Non-overlap Network

In this paper, the input feature $\boldsymbol{x} \in \mathbb{R}^n$ is i.i.d. Gaussian random vector with zero mean and unit variance. Let $\mathcal{G}$ denote this distribution. We assume that

there exists a ground truth $\boldsymbol{w}^*$ by which the training data is generated. The population risk is then:

$$f(\boldsymbol{w}) = \mathbb{E}_{\mathcal{G}}[(L(\boldsymbol{x}; \boldsymbol{w}) - L(\boldsymbol{x}; \boldsymbol{w}^*))^2]. \tag{3}$$

We define

$$g(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}_{\mathcal{G}}[\sigma(\boldsymbol{u} \cdot \boldsymbol{x})\sigma(\boldsymbol{v} \cdot \boldsymbol{x})]. \tag{4}$$

Then:

Lemma 1 *[2]. Assume $\boldsymbol{x} \in \mathbb{R}^d$ is a vector where the entries are i.i.d. Gaussian random variables with mean 0 and variance 1. Given $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$, denote by $\theta_{\boldsymbol{u},\boldsymbol{v}}$ the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$. Then*

$$g(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2\pi}\|\boldsymbol{u}\|\|\boldsymbol{v}\| \left(\sin\theta_{\boldsymbol{u},\boldsymbol{v}} + (\pi - \theta_{\boldsymbol{u},\boldsymbol{v}})\cos\theta_{\boldsymbol{u},\boldsymbol{v}}\right).$$

For the no-overlap network, the loss function is simplified to:

$$f(\boldsymbol{w}) = \frac{1}{k^2}\left[a(\|\boldsymbol{w}\|^2 + \|\boldsymbol{w}^*\|^2) - 2\,kg(\boldsymbol{w}, \boldsymbol{w}^*) - 2b\|\boldsymbol{w}\|\|\boldsymbol{w}^*\|\right]. \tag{5}$$

where $b = \frac{k^2-k}{2\pi}$ and $a = b + \frac{k}{2}$.

3.2 The Relaxed Variables Splitting Method

Let $\eta > 0$ denote the step size. Consider a simple gradient descent update:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta\nabla f(\boldsymbol{w}^t). \tag{6}$$

It was shown [1] that the one-layer non-overlap network can be learned with high probability and in polynomial time. We seek to improve sparsity in the limit weight while also maintain good accuracy. A classical method to accomplish this task is to introduce ℓ_1 regularization to the population loss function, and the modified gradient update rule. Consider the minimization problem:

$$l(\boldsymbol{w}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1. \tag{7}$$

for some $\lambda > 0$. We propose a new approach to solve this minimization problem, called the Relaxed Variable Splitting Method (RVSM). We first convert (7) into an equation of two variables

$$l(\boldsymbol{w}, \boldsymbol{u}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{u}\|_1.$$

and consider the augmented Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{u}\|_1 + \frac{\beta}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{8}$$

Let $S_{\lambda/\beta}(\boldsymbol{w}) := sgn(\boldsymbol{w})(|\boldsymbol{w}| - \lambda/\beta)\chi_{\{|\boldsymbol{w}|>\lambda/\beta\}}$ be the soft thresholding operator. The RSVM is:

Algorithm 1. RVSM

Input: η, β, λ, max_{epoch}, max_{batch}
Initialization: $\boldsymbol{w}^0$
Define: $\boldsymbol{u}^0 = S_{\lambda/\beta}(\boldsymbol{w}^0)$
for $t = 0, 1, 2, ..., max_{epoch}$ **do**
  for $batch = 1, 2, ..., max_{batch}$ **do**
    $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \nabla f(\boldsymbol{w}^t) - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^t)$
    $\boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}) = S_{\lambda/\beta}(\boldsymbol{w}^t)$
  end for
end for

3.3 Comparison with ADMM

A well-known, modern method to solve the minimization problem (7) is the Alternating Direction Method of Multipliers (or ADMM). In ADMM, we consider the Lagrangian

$$\mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{z}) = f(\boldsymbol{w}) + \lambda\|\boldsymbol{u}\|_1 + \langle \boldsymbol{z}, \boldsymbol{w} - \boldsymbol{u}\rangle + \frac{\beta}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2. \tag{9}$$

and apply the updates:

$$\begin{cases} \boldsymbol{w}^{t+1} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}, \boldsymbol{u}^t, \boldsymbol{z}^t) \\ \boldsymbol{u}^{t+1} \leftarrow \arg\min_{\boldsymbol{u}} \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}, \boldsymbol{z}^t) \\ \boldsymbol{z}^{t+1} \leftarrow \boldsymbol{z}^t + \beta(\boldsymbol{w}^{t+1} - \boldsymbol{u}^{t+1}) \end{cases} \tag{10}$$

Although widely used in practice, the ADMM method has several drawbacks when it comes to regularizing deep neural networks: First, the loss function f is often non-convex and only differentiable in some very specific regions, thus the current theory of optimizations does not apply [25]. Secondly, the update

$$\boldsymbol{w}^{t+1} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}, \boldsymbol{z}^t)$$

is not applicable in practice on DNN, as it requires one to know fully how $f(\boldsymbol{w})$ behaves. In most ADMM adaptations on DNN, this step is replaced by a simple gradient descent. Lastly, the Lagrange multiplier $\boldsymbol{z}^t$ tends to reduce the sparsity of the limit of $\boldsymbol{u}^t$, as it seeks to close the gap between $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$. In contrast, the RVSM method resolves all these difficulties presented by ADMM. First, we will show that in a one-layer non-overlap network, the iterations will keep $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$ in a nice region, where one can guarantee Lipschitz gradient property for $f(\boldsymbol{w})$. Secondly, the update of $\boldsymbol{w}^t$ is not an arg min update, but rather a gradient descent iteration itself, so our theory does not deviate from practice. Lastly, without the Lagrange multiplier term $\boldsymbol{z}^t$, there will be a gap between $\boldsymbol{w}^t$ and $\boldsymbol{u}^t$ at the limit. The $\boldsymbol{u}^t$ is much more sparse than in the case of ADMM, and numerical results showed that $f(\boldsymbol{w}^t)$ and $f(\boldsymbol{u}^t)$ behave very similarly on deep networks. An intuitive explanation for this is that when the dimension of $\boldsymbol{w}^t$ is

high, most of its components that will be pruned off to get $\boldsymbol{u}^t$ have very small magnitudes, and are often the redundant weights.

In short, the RVSM method is easier to implement (no need to keep track of the variable $\boldsymbol{z}^t$), can greatly increase sparsity in the weight variable $\boldsymbol{u}^t$, while also maintaining the same performance as ADMM. Moreover, RVSM has convergence guarantee and limit characterization as stated below.

4 Main Results

Before we state our main results, the following Lemma is needed to establish the existence of a Lipschitz constant L:

Lemma 2. *(Lipschitz gradient)*
There exists a global constant L such that the iterations of Algorithm 1 satisfy

$$\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^{t+1})\| \leq L\|\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\|, \quad \forall t. \tag{11}$$

An important consequence of Lemma 2 is: for all t, the iterations of Algorithm 1 satisfy:

$$f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) \leq \langle \nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2.$$

Theorem 1. *Suppose the initialization of the RVSM Algorithm satisfies:*

(i) Step size η is small so that $\eta \leq \frac{1}{\beta+L}$;
(ii) Initial angle $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, for some $\delta > 0$;
(iii) Parameters k, β, λ are such that $k \geq 2, \beta \leq \frac{\delta \sin \delta}{k\pi}$, and $\frac{\lambda}{\beta} < \frac{1}{\sqrt{d}}$.

Then the Lagrangian $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ decreases monotonically; and $(\boldsymbol{w}^t, \boldsymbol{u}^t)$ converges sub-sequentially to a limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$, with $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$, such that:

(i) $0 \in \partial_{\boldsymbol{u}} \mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ and $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}) = 0$;
(ii) $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t) = O(\epsilon)$ in $O(1/\epsilon^2)$ iterations;
(iii) The limit point $\bar{\boldsymbol{w}}$ is close to the ground truth $\boldsymbol{w}^*$ in the sense that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$ and $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\| = O(\beta)$.

The full proof of Theorem 1 is given in the next section. Here we overview the key steps. First, we show that the iterations of Algorithm 1 will eventually bring $\boldsymbol{w}^t$ to within a closed annulus D of width $2\,M$ around the sphere centered at origin with radius $\|\boldsymbol{w}^*\|$. In other words, there exists a T such that for all $t \geq T, \|\boldsymbol{w}^t\| \in [\|\boldsymbol{w}^*\| - M, \|\boldsymbol{w}^*\| + M]$, for some $0 < M < \|\boldsymbol{w}^*\|$. Then with no loss of generality, we can assume that $\boldsymbol{w}^t$ is in this closed strip, for all t.

Next, for the region D of the iterations, we will show there exists a global constant L such that the Lipschitz gradient property in Lemma 2 holds.

Finally, the Lipschitz gradient property allows us to show the descent of angle θ^t and Lagrangian $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$. The conditions on η, β, λ are used to show $\theta^{t+1} \leq \theta^t$; and an analysis of the limit point gives the bound on $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$ and $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$.

From the descent property of $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$, classical results from optimization [1] can be used to show that after $T = O\left(\frac{1}{\epsilon^2}\right)$ iterations, we have $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t) = O(\epsilon)$, for some $t \in (0, T]$. This leads to the desired convergence rate and finishes the proof.

It should be noted that without the condition on β being small, one may not guarantee monotonicity of θ^t. However, it still can be shown that $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ decreases and thus the iteration will converge to some limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. In this case, the limit point may not be near the ground truth $\boldsymbol{w}^*$; i.e. we may not have $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. Furthermore, the bound on $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$ will also be weaker.

Corollary 1. *Suppose the initialization of the RVSM Algorithm satisfies Theorem 1, then the $\bar{\boldsymbol{w}}$ equation below holds:*

$$\boldsymbol{w}^* = \frac{k\pi}{\pi - \theta} \beta(\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})) + C\bar{\boldsymbol{w}}, \tag{12}$$

where $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$, constant $C \in (0, \frac{1}{1-2k\lambda\sqrt{d}})$. Since component-wise, $\bar{\boldsymbol{w}} - S_{\lambda/\beta}(\bar{\boldsymbol{w}})$ has the same sign as $\bar{\boldsymbol{w}}$, the ground truth $\boldsymbol{w}^*$ is an expansion of $C\,\bar{\boldsymbol{w}}$, or equivalently $\bar{\boldsymbol{w}}$ is (up to scalar multiple) a shrinkage of $\boldsymbol{w}^*$.

The proofs of Theorem 1 and Corollary 1.1 do not require convexity of the regularization term $\lambda\|\boldsymbol{u}\|_1$, hence extend to other sparse penalties such as ℓ_0 and transformed ℓ_1 penalty [27]. We have:

Corollary 2. *Under the conditions of Theorem 1 however with the l_1 penalty replaced by an ℓ_0 or transformed-ℓ_1 penalty, the RVSM Algorithm converges sub-sequentially to a limit point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$ satisfying $\nabla_{\boldsymbol{w}} \mathcal{L}_\beta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}) = 0$. The Lagrangian and angle θ^t also decrease monotonically, with the limit angle satisfying $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) < \delta$. Here $\bar{\boldsymbol{u}}$ is a thresholding of $\bar{\boldsymbol{w}}$, and equation (12) holds with $S_{\lambda/\beta}(\cdot)$ replaced by the thresholding operator of the corresponding penalty.*

5 Proof of Main Results

The following Lemmas will be needed to prove Theorem 1:

Lemma 3. *(Properties of the gradient, [1])*
For the loss function $f(\boldsymbol{w})$ of Eq. (5), the following holds:

1. *$f(\boldsymbol{w})$ is differentiable if and only if $\boldsymbol{w} \neq 0$.*
2. *For $k > 1$, $f(\boldsymbol{w})$ has three critical points:*

(a) *A local maximum at $\boldsymbol{w} = 0$.*
(b) *A unique global minimum at $\boldsymbol{w} = \boldsymbol{w}^*$.*
(c) *A degenerate saddle point at $\boldsymbol{w} = -\left(\frac{k^2-k}{k^2+(\pi-1)k}\right)\boldsymbol{w}^*$.*

For $k = 1, w = 0$ is not a local maximum and the unique global minimum $\boldsymbol{w}^*$ is the only differentiable critical point.
Given $\theta := \theta(\boldsymbol{w}, \boldsymbol{w}^*)$, the gradient of f can be expressed as

$$\nabla f(\boldsymbol{w}) = \frac{1}{k^2}\left[\left(k + \frac{k^2-k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\sin\theta - \frac{k^2-k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\right)\boldsymbol{w} - \frac{k}{\pi}(\pi-\theta)\boldsymbol{w}^*\right]. \tag{13}$$

Lemma 4. *(Lipschitz gradient with co-planar assumption, [1])*
Assume $\|\boldsymbol{w}_1\|, \|\boldsymbol{w}_2\| \geq M$, $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}^*$ are on the same two dimensional half-plane defined by $\boldsymbol{w}^*$, then

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\| \leq L\|\boldsymbol{w}_1 - w_2\|$$

for $L = 1 + \frac{3\|\boldsymbol{w}^*\|}{M}$.

Lemma 5. *For $k \geq 1$, there exists constants $M_k, T > 0$ such that for all $t \geq T$, the iterations of Algorithm 1 satisfy:*

$$\|\boldsymbol{w}^t\| \in [\|\boldsymbol{w}^*\| - M_k, \|\boldsymbol{w}^*\| + M_k]. \tag{14}$$

where $M_k < \|\boldsymbol{w}^*\|$, and $M_k \to 0$ as $k \to \infty$.

From Lemma 5, WLOG, we will assume that $T = 0$.

Lemma 6. *(Descent of $\mathcal{L}_\beta$ due to $\boldsymbol{w}$ update)*
For η small such that $\eta \leq \frac{1}{\beta+L}$, we have

$$\mathcal{L}_\beta(\boldsymbol{u}^{t+1}, \boldsymbol{w}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t).$$

5.1 Proof of Lemma 2

By Algorithm 1 and Lemma 5, $\|\boldsymbol{w}^t\| \geq \|\boldsymbol{w}^*\| - M > 0$, for all t, and $\boldsymbol{w}^{t+1}$ is in some closed neighborhood of $\boldsymbol{w}^t$. We consider the subspace spanned by $\boldsymbol{w}^t, \boldsymbol{w}^{t+1}$, and $\boldsymbol{w}^*$, this reduces the problem to a 3-dimensional space.

Consider the plane formed by $\boldsymbol{w}^t$ and $\boldsymbol{w}^*$. Let $\boldsymbol{v}^{t+1}$ be the point on this plane, closest to $\boldsymbol{w}^t$, such that $\|\boldsymbol{w}^{t+1}\| = \|\boldsymbol{v}^{t+1}\|$ and $\theta(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*) = \theta(\boldsymbol{v}^{t+1}, \boldsymbol{w}^*)$. In other words, $\boldsymbol{v}^{t+1}$ is the intersection of the plane formed by $\boldsymbol{w}^t, \boldsymbol{w}^*$ and the cone with tip at zero, side length $\|\boldsymbol{w}^{t+1}\|$, and main axis $\boldsymbol{w}^*$ (See Fig. 2). Then

$$\begin{aligned}
&\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^{t+1})\| \\
\leq &\|\nabla f(\boldsymbol{w}^t) - \nabla f(\boldsymbol{v}^{t+1})\| + \|\nabla f(\boldsymbol{v}^{t+1}) - \nabla f(\boldsymbol{w}^{t+1})\| \\
\leq &L_1\|\boldsymbol{w}^t - \boldsymbol{v}^{t+1}\| + L_2\|\boldsymbol{v}^{t+1} - \boldsymbol{w}^{t+1}\|
\end{aligned} \tag{15}$$

for some constants L_1, L_2. The first term is bounded since $\boldsymbol{w}^t, \boldsymbol{v}^{t+1}, \boldsymbol{w}^*$ are co-planar by construction, and Lemma 4 applies. The second term is bounded

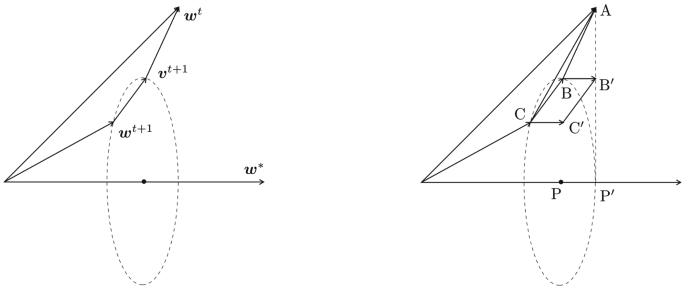


Fig. 2. Geometry of the update of $\boldsymbol{w}^t$ and the corresponding $\boldsymbol{w}^{t+1}, \boldsymbol{v}^{t+1}$.

by applying Eq. 13 with $\|\boldsymbol{w}^{t+1}\| = \|\boldsymbol{v}^{t+1}\|$ and $\theta(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*) = \theta(\boldsymbol{v}^{t+1}, \boldsymbol{w}^*)$. It remains to show there exists a constant $L_3 > 0$ such that

$$\|\boldsymbol{w}^t - \boldsymbol{v}^{t+1}\| + \|\boldsymbol{v}^{t+1} - \boldsymbol{w}^{t+1}\| \leq L_3 \|\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\|$$

Let A, B, C be the tips of $\boldsymbol{w}^t, \boldsymbol{v}^{t+1}, \boldsymbol{w}^{t+1}$, respectively. Let P be the point on $\boldsymbol{w}^*$ that is at the base of the cone (so P is the center of the circle with B, C on the arc). We will show there exists a constant L_3 such that

$$|AB| + |BC| \leq L_3 |AC| \tag{16}$$

Case 1: A, B, P are collinear: By looking at the cross-section of the plane formed by AB, AC, it can be seen that AC is not the smallest edge in $\triangle ABC$. Thus there exists some L_3 such that Eq. 16 holds.
Case 2: A, B, P are not collinear: Translate B, C, P to B', C', P' such that A, B', P' are collinear and $BB', CC', PP' /\!/ \boldsymbol{w}^*$. Then by Case 1:

$$|AB'| + |B'C'| \leq L_3 |AC'|$$

and AC' is not the smallest edge in $\triangle AB'C'$. By back-translating B', C' to B, C, it can be seen that AC is again not the smallest edge in $\triangle ABC$. This implies

$$|AB| + |BC| \leq L_4 |AC|$$

for some constant L_4. Thus Eq. 16 is proved. Combining with Eq. 15, Lemma 2 is proved.

5.2 Proof of Lemma 5

First we show that if $\|\boldsymbol{w}^t\| < \|\boldsymbol{w}^*\|$, then the update of Algorithm 1 will satisfy $\|\boldsymbol{w}^{t+1}\| > \|\boldsymbol{w}^t\|$. By Lemma 3,

$$\begin{aligned}\nabla f(\boldsymbol{w}) =& \frac{1}{k^2}\left[\left(k + \frac{k^2 - k}{\pi} - \frac{k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\sin\theta - \frac{k^2 - k}{\pi}\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}\|}\right)\boldsymbol{w} - \frac{k}{\pi}(\pi - \theta)\boldsymbol{w}^*\right] \\ =& \frac{1}{k^2}(C_1 \boldsymbol{w} - C_2 \boldsymbol{w}^*)\end{aligned}$$

so the update of $\boldsymbol{w}^t$ reads

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \frac{C_1^t + \beta k^2}{k^2} \boldsymbol{w}^t + \eta \frac{C_2^t}{k^2} \boldsymbol{w}^* + \eta\beta \boldsymbol{u}^{t+1},$$

where $C_2^t > 0$. Since $\boldsymbol{u}^{t+1} = S_{\lambda/\beta}(\boldsymbol{w}^t)$, the term $\eta\beta\boldsymbol{u}^{t+1}$ will increase the norm of $\boldsymbol{w}^t$. For the remaining terms,

$$\begin{aligned} C_1^t &= k + \frac{k^2 - k}{\pi} - \frac{k}{\pi} \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} \sin\theta - \frac{k^2 - k}{\pi} \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} \\ &\leq k + \frac{k^2 - k}{\pi} \left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) \end{aligned}$$

When $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$ is large, C_1^t is negative. The update will increase the norm of $\|\boldsymbol{w}^t\|$ if $C_1^t + \beta k^2 \leq 0$ and

$$\left\| \frac{C_1^t + \beta k^2}{k^2} \boldsymbol{w}^t \right\| > \left\| \frac{C_2^t}{k^2} \boldsymbol{w}^* \right\|$$

This condition is satisfied when

$$-\left[k + \frac{k^2 - k}{\pi} \left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) + \beta k^2\right] > \frac{k}{\pi} \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$

When $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} > 1$, the LHS is $O(k^2)$, while the RHS is $O(k)$. Thus there exists some M_k such that $\boldsymbol{w}^t$ will eventually stay in the region $\|\boldsymbol{w}^t\| \geq \|\boldsymbol{w}^*\| - M_k$. Moreover, as $k \to \infty$, we have $M_k \to 0$.
Next, when $\|\boldsymbol{w}^t\| > \|\boldsymbol{w}^*\|$, the update of $\boldsymbol{w}^t$ reads

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \frac{C_1^t}{k^2} \boldsymbol{w}^t + \eta \frac{C_2^t}{k^2} \boldsymbol{w}^* - \eta\beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

the last term decreases the norm of $\boldsymbol{w}^t$. In this case, C_1^t is positive and

$$C_1^t \geq \frac{k\pi - k}{\pi} + \frac{k^2 - k}{\pi} \left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right)$$

The update will decrease the norm of $\boldsymbol{w}^t$ if

$$\frac{k\pi - k}{\pi} + \frac{k^2 - k}{\pi} \left(1 - \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}\right) > \frac{k}{\pi} \frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|}$$

which holds when $\frac{\|\boldsymbol{w}^*\|}{\|\boldsymbol{w}^t\|} < 1$, and the Lemma is proved.

5.3 Proof of Lemma 6

Proof. By the update of $\boldsymbol{u}^t$, $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^{t+1}) \leq \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$. For the update of $\boldsymbol{w}^t$, notice that

$$\nabla f(\boldsymbol{w}^t) = \frac{1}{\eta} \left(\boldsymbol{w}^t - \boldsymbol{w}^{t+1}\right) - \beta(\boldsymbol{w}^t - \boldsymbol{u}^{t+1})$$

Then for a fixed $\boldsymbol{u} := \boldsymbol{u}^{t+1}$, we have

$$\begin{aligned}
&\mathcal{L}_\beta(\boldsymbol{w}^{t+1}, \boldsymbol{u}) - \mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}) \\
=&f(\boldsymbol{w}^{t+1}) - f(\boldsymbol{w}^t) + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
\leq&\langle \nabla f(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
+&\frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
=&\frac{1}{\eta}\langle \boldsymbol{w}^t - \boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \beta\langle \boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle \\
+&\frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 + \frac{\beta}{2}\left(\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \|\boldsymbol{w}^t - \boldsymbol{u}\|^2\right) \\
=&\frac{1}{\eta}\langle \boldsymbol{w}^t - \boldsymbol{w}^{t+1}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle + \left(\frac{L}{2} + \frac{\beta}{2}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
+&\frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{u}\|^2 - \frac{\beta}{2}\|\boldsymbol{w}^t - \boldsymbol{u}\|^2 \\
-&\beta\langle \boldsymbol{w}^t - \boldsymbol{u}, \boldsymbol{w}^{t+1} - \boldsymbol{w}^t\rangle - \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2 \\
=&\left(\frac{L}{2} + \frac{\beta}{2} - \frac{1}{\eta}\right)\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2
\end{aligned}$$

Therefore, if η is small so that $\eta \leq \frac{2}{\beta + L}$, the update on $\boldsymbol{w}$ will decrease $\mathcal{L}_\beta$.

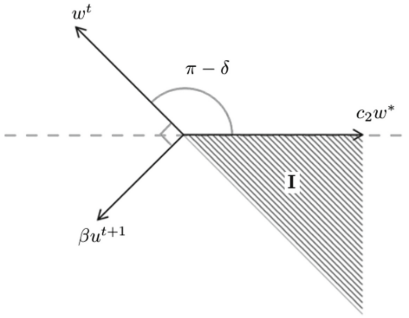


Fig. 3. Worst case of the update on $\boldsymbol{w}^t$

5.4 Proof of Theorem 1

We will first show that if $\theta(\boldsymbol{w}^0, \boldsymbol{w}^*) \leq \pi - \delta$, then $\theta(\boldsymbol{w}^t, \boldsymbol{w}^*) \leq \pi - \delta$, for all t. We will show $\theta(\boldsymbol{w}^1, \boldsymbol{w}^*) \leq \pi - \delta$, the statement is then followed by induction.

To this end, by the update of $\boldsymbol{w}^t$, one has

$$= C\boldsymbol{w}^0 + \eta\frac{\pi - \theta(\boldsymbol{w}^0, \boldsymbol{w}^*)}{k\pi}\boldsymbol{w}^* + \eta\beta\boldsymbol{u}^1$$

for some constant $C > 0$. Since $\boldsymbol{u}^1 = S_{\lambda/\beta}(\boldsymbol{w}^0), \theta(\boldsymbol{u}^1, \boldsymbol{w}^0) \leq \frac{\pi}{2}$. Notice that the sum of the first two terms on the RHS brings the vector closer to $\boldsymbol{w}^*$, while the last term may behave unexpectedly. Consider the worst case scenario: $\boldsymbol{w}^0, \boldsymbol{w}^*, \boldsymbol{u}^1$ are co-planar with $\theta(\boldsymbol{u}^1, \boldsymbol{w}^0) = \frac{\pi}{2}$, and $\boldsymbol{w}^*, \boldsymbol{u}^1$ are on two sides of $\boldsymbol{w}^0$ (See Fig. 3). We need $\frac{\delta}{k\pi}\boldsymbol{w}^* + \beta\boldsymbol{u}^1$ to be in region I. This condition is satisfied when β is small such that

$$\sin\delta \geq \frac{\beta\|\boldsymbol{u}^1\|}{\frac{\delta}{k\pi}\|\boldsymbol{w}^*\|} = \frac{k\pi\beta\|\boldsymbol{u}^1\|}{\delta}$$

since $\|\boldsymbol{u}^1\| \leq 1$, it is sufficient to have $\beta \leq \frac{\delta\sin\delta}{k\pi}$.

Next, consider the limit of the RVSM algorithm. Since $\mathcal{L}_\beta(\boldsymbol{w}^t, \boldsymbol{u}^t)$ is non-negative, by Lemma 6, $\mathcal{L}_\beta$ converges to some limit $\mathcal{L}$. This implies $(\boldsymbol{w}^t, \boldsymbol{u}^t)$ converges to some stationary point $(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. By Lemma 3 and the update of $\boldsymbol{w}^t$, we have

$$\overline{\boldsymbol{w}} = c_1\overline{\boldsymbol{w}} + \eta c_2\boldsymbol{w}^* + \eta\beta\overline{\boldsymbol{u}} \tag{17}$$

for some constant $c_1 > 0, c_2 \geq 0$, where $c_2 = \frac{\pi-\theta}{k\pi}$, with $\theta := \theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*)$, and $\bar{\boldsymbol{u}} = S_{\lambda/\beta}(\bar{\boldsymbol{w}})$. If $c_2 = 0$, then we must have $\bar{\boldsymbol{w}} /\!/ \bar{\boldsymbol{u}}$. But since $\bar{\boldsymbol{u}} = S_{\lambda/\beta}$, this implies all non-zero components of $\bar{\boldsymbol{w}}$ are either equal in magnitude, or all have magnitude smaller than $\frac{\lambda}{\beta}$. The latter case is not possible when $\frac{\lambda}{\beta} < \frac{1}{\sqrt{d}}$. Furthermore, $c_2 = 0$ when $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) = \pi$ or 0. We have shown that $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) \leq \pi - \delta$, thus $\theta(\bar{\boldsymbol{w}}, \boldsymbol{w}^*) = 0$. Thus, $\bar{\boldsymbol{w}} = \boldsymbol{w}^*$, and all non-zero components of $\boldsymbol{w}^*$ are equal in magnitude. This has probability zero if we assume $\boldsymbol{w}^*$ is initiated uniformly on the unit circle. Hence we will assume that almost surely, $c_2 > 0$. Expression (17) implies

$$c_2\boldsymbol{w}^* + \beta\bar{\boldsymbol{u}} /\!/ \bar{\boldsymbol{w}} \tag{18}$$

Expression (18) implies $\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}}$, and $\boldsymbol{w}^*$ are co-planar. Let $\gamma := \theta(\bar{\boldsymbol{w}}, \bar{\boldsymbol{u}})$. From expression (18), and the assumption that $\|\boldsymbol{w}^*\| = 1$, we have

$$(\langle c_2\boldsymbol{w}^* + \beta\bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}\rangle)^2 = \|c_2\boldsymbol{w}^* + \beta\bar{\boldsymbol{u}}\|^2\|\bar{\boldsymbol{w}}\|^2$$

or

$$\begin{aligned}&\|\bar{\boldsymbol{w}}\|^2(c_2^2\cos^2\theta + 2c_2\beta\|\bar{\boldsymbol{u}}\|\cos\theta\cos\gamma + \beta^2\|\bar{\boldsymbol{u}}\|^2\cos^2\gamma)\\ =&\|\bar{\boldsymbol{w}}\|^2(c_2^2 + 2c_2\beta\|\bar{\boldsymbol{u}}\|\cos(\theta+\gamma) + \beta^2\|\bar{\boldsymbol{u}}\|^2)\end{aligned}$$

This reduces to

$$c_2^2\sin^2\theta - 2c_2\beta\|\bar{\boldsymbol{u}}\|\sin\theta\sin\gamma + \beta^2\|\bar{\boldsymbol{u}}\|^2\sin^2\gamma = 0,$$

which implies $\frac{\pi-\theta}{k\pi}\sin\theta = \beta\|\bar{\boldsymbol{u}}\|\sin\gamma$. By the initialization $\beta \leq \frac{\delta\sin\delta}{k\pi}$, we have $\frac{\pi-\theta}{k\pi}\sin\theta < \frac{\delta}{k\pi}\sin\delta$. This implies $\theta < \delta$.

Finally, the limit point satisfies $\|\nabla f(\bar{\boldsymbol{w}})+\beta(\bar{\boldsymbol{w}}-\bar{\boldsymbol{u}})\| = 0$. By the initialization requirement, we have $\|\beta(\bar{\boldsymbol{w}} - \bar{\boldsymbol{u}})\| < \beta \leq \frac{\delta \sin \delta}{k\pi}$. This implies $\|\nabla f(\bar{\boldsymbol{w}})\| \leq \frac{\delta \sin \delta}{k\pi}$. By the Lipschitz gardient property in Lemma 2 and critical points property in Lemma 3, $\bar{\boldsymbol{w}}$ must be close to $\boldsymbol{w}^*$. In other words, $\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\|$ is comparable to the chord length of the circle of radius $\|\boldsymbol{w}^*\|$ and angle θ:

$$\begin{aligned}\|\bar{\boldsymbol{w}} - \boldsymbol{w}^*\| &= O\left(2\sin\left(\frac{\theta}{2}\right)\right) = O(\sin\theta)\\ &= O\left(\frac{k\pi\beta\|\bar{\boldsymbol{u}}\|\sin\gamma}{\pi-\theta}\right) = O(k\beta\sin\gamma).\end{aligned}$$

6 Numerical Experiments

First, we experiment RVSM with VGG-16 on the CIFAR10 data set. Table 1 shows the result of RVSM under different penalties. The parameters used are $\lambda = 1.e-5, \beta = 1.e-2$, and $a = 1$ for Tℓ_1 penalty. It can be seen that RVSM can maintain very good accuracy while also promotes good sparsity in the trained network. Between the penalties, ℓ_0 gives the best sparsity, ℓ_1 the best accuracy, and Tℓ_1 gives a middle ground between ℓ_0 and ℓ_1. Since the only difference between these parameters is in the pruning threshold, in practice, one may simply stick to ℓ_0 regularization and just fine-tune the hyper-parameters.

Secondly, we experiment our method on ResNet18 and the CIFAR10 data set. The results are displayed in Table 2. The base model was trained on 200 epochs using standard SGD method with initial learning rate 0.1, which decays by a factor of 10 at the 80th, 120th, and 160th epochs. For the RVSM method, we use ℓ_0 regularization and set λ =1.e-6, β = 8.e-2. For ADMM, we set the pruning threshold to be 60% and ρ =1.e-2. The ADMM method implemented here is per [28], an "empirical variation" of the true ADMM (Eq. 10). In particular, the arg min update of $\boldsymbol{w}^t$ is replaced by a gradient descent step. Such "modified" ADMM is commonly used in practice on DNN.

It can be seen in Table 2 that RVSM runs quite effectively on the benchmark deep network, promote much better sparsity than ADMM (93.70% vs. 47.08%), and has slightly better performance. The sparsity here is the percentage of zero components over all network weights.

Table 1. Sparsity and accuracy of RVSM under different penalties on VGG-16 on CIFAR10.

Penalty	Accuracy	Sparsity
Base model	93.82	0
ℓ_1	93.7	35.68
Tℓ_1	93.07	63.34
ℓ_0	92.54	86.89

Table 2. Comparison between ADMM and RVSM (ℓ_0) for ResNet18 training on the CIFAR10 dataset.

ResNet18	Accuracy	Sparsity
SGD	95.07	0
ADMM	94.84	47.08
RVSM (ℓ_0)	94.89	93.70

7 Conclusion

We proved the global convergence of RVSM to sparsify a convolutional ReLU network on a regression problem and analyzed the sparsity of the limiting weight vector as well as its error estimate from the ground truth (i.e. the global minimum). The proof used geometric argument to establish angle and Lagrangian descent properties of the iterations thereby overcame the non-existence of gradient at the origin of the loss function. Our experimental results provided additional support for the effectiveness of RVSM via ℓ_0, ℓ_1 and Tℓ_1 penalties on standard deep networks and CIFAR-10 image data. In future work, we plan to extend RVSM theory to multi-layer network and structured (channel/filter/etc.) pruning.

Acknowledgments. The authors would like to thank Dr. Yingyong Qi for suggesting reference [12], and Dr. Penghang Yin for helpful discussions. The work was partially supported by NSF grants DMS-1522383, IIS-1632935, and DMS-1854434.

References

1. Brutzkus, A., Globerson, A.: Globally optimal gradient descent for a convnet with gaussian inputs (2017). ArXiv preprint 1702.07966
2. Cho, Y., Saul, L.K.: Kernel methods for deep learning. In: Advances in neural information processing systems, pp. 342–350 (2009)
3. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks (2016). ArXiv preprint 1612.08083
4. Du, S., Lee, J., Tian, Y.: When is a convolutional filter easy to learn? (2017). ArXiv 1709.06129
5. Du, S., Lee, J., Tian, Y., Poczos, B., Singh, A.: Gradient descent learns one-hidden-layer CNN: don't be afraid of spurious local minima. In: International Conference on Machine Learning, ICML (2018)
6. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
7. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding (2015). ArXiv preprint 1510.00149
8. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)

9. Kingma, D., Ba, J.: Adam: a method for stochastic optimization (2014). ArXiv preprint 1412.6980
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
11. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. In: NIPS, vol. 2, pp. 598–605 (1989)
12. Louizos, C., Welling, M., Kingma, D.: Learning sparse neural networks through ℓ_0 regularization (2018). ArXiv preprint 1712.01312v2
13. Lu, Z., Zhang, Y.: Penalty decomposition methods for rank minimization. Optim. Methods Softw. **30**(3), 531–558 (2014). https://doi.org/10.1080/10556788.2014.936438
14. Molchanov, D., Ashukha, A., Vetrov, D.: Variational dropout sparsifies deep neural networks (2017). ArXiv preprint 1701.05369
15. Nikolova, M.: Local strong homogeneity of a regularized estimator. SIAM J. Appl. Math. **61**(2), 633–658 (2000)
16. Polyak, B.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)
17. Reddi, S., Kale, S., Kumar, S.: On the convergence of adam and beyond. In: International Conference on Learning Representations (2018)
18. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951)
19. Rumelhart, D., Hinton, G., Williams, R.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
20. Shamir, O.: Distribution-specific hardness of learning neural networks (2016). ArXiv preprint 1609.01037
21. Taylor, G., Burmeister, R., Xu, Z., Singh, B., Patel, A., Goldstein, T.: Training neural networks without gradients: a scalable admm approach. In: International Conference on Machine Learning, pp. 2722–2731 (2016)
22. Tian, Y.: An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis (2017). ArXiv preprint 1703.00560
23. Tieleman, T., Hinton, G.: Divide the gradient by a running average of its recent magnitude. coursera: neural networks for machine learning. Technical report (2017)
24. Ullrich, K., Meeds, E., Welling, M.: Soft weight-sharing for neural network compression. In: ICLR (2017)
25. Wang, Y., Zeng, J., Yin, W.: Global convergence of ADMM in nonconvex nonsmooth optimization. J. Sci. Comput. **78**(1), 29–63 (2018). https://doi.org/10.1007/s10915-018-0757-z
26. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization (2016). ArXiv preprint 1611.03530
27. Zhang, S., Xin, J.: Minimization of transformed l_1 penalty: closed form representation and iterative thresholding algorithms. Commun. Math. Sci. **15**(2), 511–537 (2017). https://doi.org/10.4310/cms.2017.v15.n2.a9
28. Zhang, T., Ye, S., Zhang, K., Tang, J., Wen, W., Fardad, M., Wang, Y.: A systematic DNN weight pruning framework using alternating direction method of multipliers. arXiv preprint 1804.03294 (2018). https://arxiv.org/abs/1804.03294

Comparison of Hybrid Recurrent Neural Networks for Univariate Time Series Forecasting

Anibal Flores(✉), Hugo Tito, and Deymor Centty

Universidad Nacional de Moquegua, Moquegua, Peru
anibalf11@gmail.com

Abstract. The work presented in this paper aims to improve the accuracy of forecasting models in univariate time series, for this it is experimented with different hybrid models of two and four layers based on recurrent neural networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). It is experimented with two time series corresponding to downward thermal infrared and all sky insolation incident on a horizontal surface obtained from NASA's repository. In the first time series, the results achieved by the two-layer hybrid models (LSTM + GRU and GRU + LSTM) outperformed the results achieved by the non-hybrid models (LSTM + LSTM and GRU + GRU); while only two of six four-layer hybrid models (GRU + LSTM + GRU + LSTM and LSTM + LSTM + GRU + GRU) outperformed non-hybrid models (LSTM + LSTM + LSTM + LSTM and GRU + GRU + GRU + GRU). In the second time series, only one model (LSTM + GRU) of two hybrid models outperformed the two non-hybrid models (LSTM + LSTM and GRU + GRU); while the four-layer hybrid models, none could exceed the results of the non-hybrid models.

Keywords: Hybrid recurrent neural networks · Long Short-Term Memory (LSTM) · Gated Recurrent Unit (GRU)

1 Introduction

The forecast of time series is one of the most interesting [1] and useful tasks in different areas of knowledge [2] such as meteorology, marketing, biology, economics, demography, etc.

The need for increasingly accurate predicted data is crucial for proper decision making: In meteorology more accurate predictions are required to be able to predict natural phenomena and mitigate negative effects, in Marketing more accurate predictions are required to be able to predict future sales of certain products or services and thus maximize profits and provide a better service to customers, and so on in each area of knowledge.

According to that indicated in the preceding paragraph, the work presented is aimed at improving the accuracy of the time series forecast models. In that sense, the implementation of hybrid prediction models is proposed as it is done in [3, 4] and [5]. However, state of the art proposals that use LSTM and GRU only use a basic two-layer architecture fundamentally LSTM + GRU. In this work, in addition to the basic architecture,

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 375–387, 2021.
https://doi.org/10.1007/978-3-030-55180-3_28

a four-layer architecture is experimented, creating different models combining LSTM layers and GRU layers in a balanced way. Table 1 shows the different hybrid models experimented in this work.

Table 1. Hybrid recurrent neural networks

Number of layers	Architecture
Two (2)	LSTM GRU
	GRU LSTM
Four (4)	LSTM LSTM GRU GRU
	GRU GRU LSTM LSTM
	LSTM GRU LSTM GRU
	GRU LSTM GRU LSTM
	LSTM GRU GRU LSTM
	GRU LSTM LSTM GRU

The time series chosen for experimentation correspond to downward thermal infrared and all sky insolation incident on a horizontal surface in the southern region of Peru, an area that throughout the year enjoys a dry and sunny climate typical of the Atacama region of South America. Figure 1 partially shows the time series mentioned.

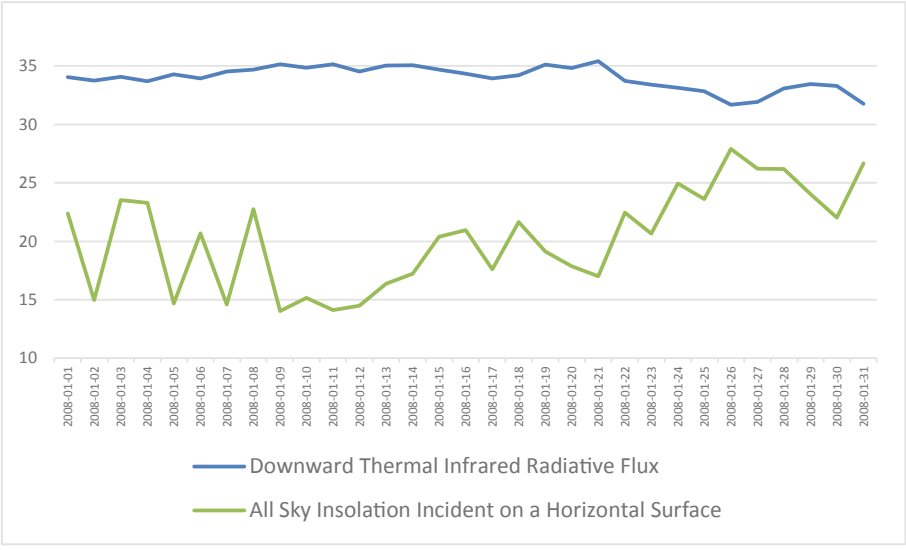


Fig. 1. Time series for experimentation.

The analysis of this type of time series would allow determining the negative effect on health of solar radiation such as skin cancer, premature aging, and cataracts, among others; likewise, it would allow determining the level of solar energy that could be obtained as an alternative to the energy obtained from fossil fuels. However, this study is limited only to the analysis of the effectiveness of hybrid forecasting models.

This work has been structured as follows: The second section shows the works related to the work presented. The third section describes certain theoretical concepts that will allow a better understanding of the paper content. In the fourth section, the process of implementing hybrid models is described. In the fifth section, the results achieved are described and discussed. In the sixth section the conclusion of the work is described and finally the future work that can be carried out from the work presented is described.

2 Related Work

Some works that show results of hybrid models are detailed below:

In [3] authors propose a hybrid model for predicting stock prices based on deep learning. The hybrid model was based on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) and it had only two layers, the first layer being LSTM and the second GRU. The results achieved show that the hybrid model exceeds the models with which it was compared: CNN, MLP, RNN, etc.

In [4] the authors proposed several hybrid models based on Singular Spectrum Analysis (SSA) Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), and Support Vector Regression (SVR) for wind speed forecasting. The results show that the hybrid model based on SSA CNN GRU and SVR in the three case studies analyzed achieved the best results.

In [5] the authors propose a hybrid model for time series classification based on Fully Convolutional Networks (FCN) and Gated Recurrent Unit (GRU). The results show that proposed hybrid model FCN GRU outperforms the results of the models with which it was compared.

In [6] the authors proposed a hybrid model for the detection and classification of volcano-seismic signals with recurrent neural networks based on vanilla and RNN. The results are compared with those achieved by the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) independently. From three study cases, GRU was better in two cases and LSTM in one case.

Some works with experimentation on solar irradiance time series are:

In [7] the authors present a review of different techniques used to forecast solar irradiance such as ARMA, ARIMA, CARDs, Artificial Neural Networks, and Wavelet Neural Network.

In [8] the authors analyze eleven statistical and machine learning techniques which are used to forecast solar irradiance. The proposed models are compared with the Root Mean Squared Error (RMSE) metric, of which Auto-Regressive Moving Average and Multi-Layer Perceptron techniques stand out as the most efficient.

In [9] the authors propose a comparison of different supervised machine learning techniques such as neural networks, Gaussian processes and vector support machines. Likewise, a simple linear autoregressive technique is implemented. The results show that machine-based techniques outperform the autoregressive linear technique.

3 Background

3.1 Deep Learning

Deep Learning is a set of machine learning algorithms that attempts to model high-level abstractions in data using computational architectures that support multiple non-linear and iterative transformations of data expressed in matrix form [10].

Deep learning algorithms contrast with shallow learning algorithms by the number of transformations applied to the signal as it propagates from the input layer to the output layer [11]. Each of these transformations includes parameters that can be trained as weights and thresholds.

3.2 Recurrent Neural Networks (RNN)

A RNN is a neural network for modeling time series [12]. The structure of this type of neural network is very similar to perceptron multilayer (MLP) with the difference that it allows connections between hidden units associated with a time delay. Through these connections, the neural network can retain information from the past [13], allowing it to discover temporal correlations between events that may be very far from each other. Figure 2 shows the typical architecture of recurrent neural network.

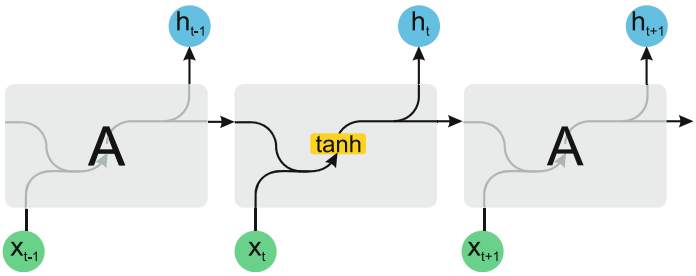


Fig. 2. Architecture of recurrent neural network

3.3 Long Short-Term Memory (LSTM)

The LSTM network was created with the goal of addressing the vanishing gradients problem, this due to the unfold process of an RNN. LSTM units allow gradients to also flow without changes, due to this, LSTM networks can still suffer the problem of gradient explosion. LSTM networks have proved to be better than conventional RNNs [14], especially when they have several layers for each time step. Figure 3 shows the typical LSTM architecture.

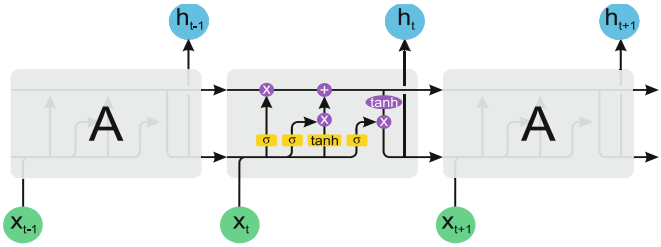


Fig. 3. Architecture of LSTM Network

3.4 Gated Recurrent Unit (GRU)

GRUs are a gating mechanism in recurrent neural networks [15]. The GRU is a recurrent neural network inspired by LSTM and it contains fewer parameters than LSTM, since it does not have an output gate. GRUs have been shown to exhibit even better performance in certain smaller datasets. However, the LSTM is stronger than the GRU, since it can easily perform an unlimited count, while the GRU cannot [16]. Figure 4 shows the architecture of Gate Recurrent (GRU) Unit network.

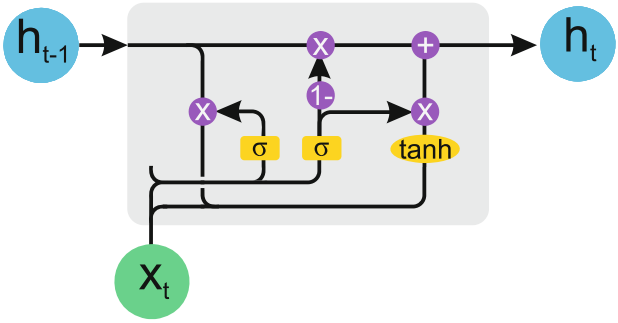


Fig. 4. Architecture of GRU Network

From Fig. 3, it is possible to get the following equations:

$$Z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h \tag{3}$$

Where:
x_t: input vector
h_t: output vector
z_t: updated gate vector
r_t: reset gate vector

W, U and b: matrix parameters and vector
σ_g: sigmoid function
σ_h: hiperbolic tangent

4 Hybrid Models

To implement the hybrid models, the following procedure was performed:

4.1 Time Series Selection

The time series chosen correspond to daily downward thermal infrared and all sky insolation incident on a horizontal surface in Moquegua city (South of Perú: Lat.-17.1898, Lon. -70.9358). Data was obtained from the NASA repository through the Power Data Access Viewer tool. For the training of hybrid models, data from 2008-01-01 to 2015-12-31 is used.

4.2 Imputation of Missing Values

The data obtained presented some missing values, which were completed through the univariate imputation technique known as Local Average of Nearest Neighbors (LANN) [17].

4.3 Architecture of Hybrid Models

The architecture of the two-layer hybrid models is similar to Fig. 5 and for four-layer hybrid models is similar to Fig. 6 [1]. These were implemented in Python language.

```
model = Sequential()

model.add(LSTM(units=60, return_sequences=True, input_shape=(lstm_ftrs.shape[1], 1)))
model.add(Dropout(0.2))

model.add(GRU(units=60))
model.add(Dropout(0.2))

model.add(Dense(units = 1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')
model.fit(lstm_ftrs, labels, epochs = 100, batch_size = 60)
```

Fig. 5. Architecture of two-layer hybrid model.

```
model = Sequential()

model.add(LSTM(units=60, return_sequences=True, input_shape=(lstm_ftrs.shape[1], 1)))
model.add(Dropout(0.2))

model.add(LSTM(units=60, return_sequences=True))
model.add(Dropout(0.2))

model.add(GRU(units=60, return_sequences=True))
model.add(Dropout(0.2))

model.add(GRU(units=60))
model.add(Dropout(0.2))

model.add(Dense(units = 1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')
model.fit(lstm_ftrs, labels, epochs = 100, batch_size = 60)
```

Fig. 6. Architecture of four-layer hybrid model.

4.4 Evaluation of Predicted Values

The results of each hybrid model are evaluated through Root Mean Squared Error (RMSE) that is calculated through Eq. (4) using testing data. Testing data correspond to the year 2016.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n-1} (Pi - Ri)^2}{n}} \tag{4}$$

Where:
n: number of predicted values
Pi: vector of predicted values
Ri: vector of real values (testing data)

The process followed for the first time series (downward thermal infrared) is repeated for the second time series (all sky insolation incident on a horizontal surface).

5 Results and Discussion

After the implementation of the different hybrid models, the following results were obtained.

According to Table 2, for downward thermal infrared time series with two-layer models on average the results of the hybrid architectures outperformed the non-hybrid ones, being the best hybrid architecture GRU LSTM (RMSE 0.4335). Likewise, it is appreciated that the GRU LSTM hybrid model in 6 of the 7 prediction cases exceeds the non-hybrid models. Only for the case of 90 predicted data was it surpassed by the non-hybrid GRU GRU model.

Table 2. Two-layers results for downward thermal infrared.

Model	Predicted days							Average
	15	30	60	90	120	150	180	
LSTM GRU	0.5000	0.4880	0.5118	0.4506	0.4728	0.4391	0.4585	0.4744
GRU LSTM	**0.4431**	**0.4170**	**0.4447**	**0.4107**	**0.4472**	**0.4301**	**0.4418**	**0.4335**
Non-hybrid architectures								
LSTM LSTM	**0.4509**	0.4766	0.5516	0.4814	**0.5102**	**0.4911**	**0.4960**	0.4939
GRU GRU	0.4710	**0.4259**	**0.4158**	**0.4812**	0.5159	0.5477	0.5828	**0.4914**

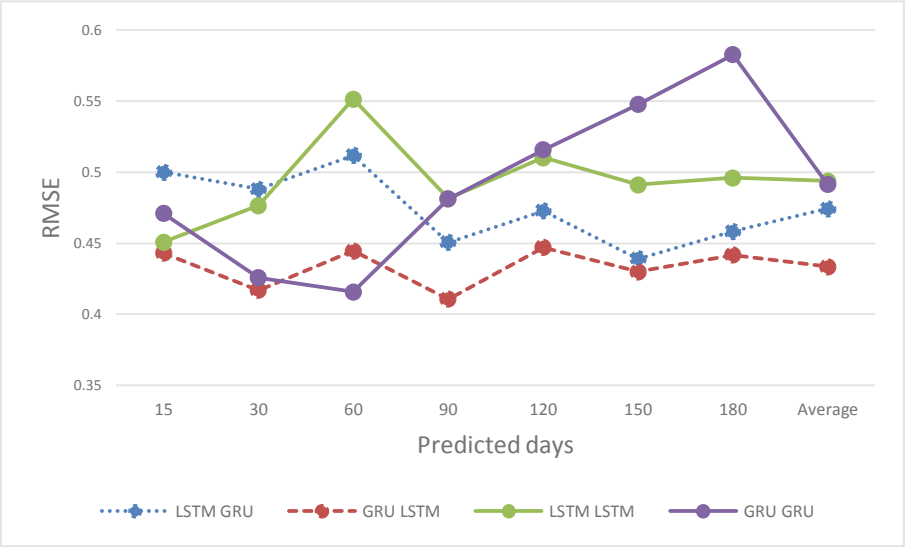


Fig. 7. Comparison between hybrid architectures and the non-hybrid.

Figure 7 shows a graphical comparison between the hybrid architectures and the non-hybrid.

According to Fig. 7, it can be seen that hybrid architectures show less variability in their RMSEs compared to non-hybrid architectures.

According to Table 3, for downward thermal infrared time series with four-layer models on average the best hybrid architecture GRU LSTM GRU LSTM (RMSE 0.4298) outperformed the non-hybrid architectures. Of the 7 prediction cases analyzed, the non-hybrid GRU LSTM architecture outperformed the non-hybrid in all cases.

Figure 8 shows a graphical comparison of the two best hybrid architectures GRU LSTM GRU LSTM (RMSE 0.4298), LSTM LSTM GRU GRU (RMSE 0.4308) and the non-hybrid.

Table 3. Four-layers results for downward thermal infrared.

Model	15	30	60	90	120	150	180	Average
LSTM LSTM GRU GRU	0.4314	0.4254	0.4511	0.4146	0.4348	**0.4177**	**0.4409**	0.4308
GRU GRU LSTM LSTM	0.5590	0.5591	0.6030	0.5219	0.5454	0.5155	0.5358	0.5485
LSTM GRU LSTM GRU	0.5033	0.4612	0.4984	0.4781	0.5224	0.5825	0.6293	0.5250
GRU LSTM GRU LSTM	**0.4268**	**0.4070**	**0.4123**	**0.4240**	**0.4470**	0.4353	0.4565	**0.4298**
LSTM GRU GRU LSTM	0.5458	0.5348	0.5824	0.5134	0.5401	0.5259	0.5542	0.5423
GRU LSTM LSTM GRU	0.4944	0.4994	0.5876	0.5108	0.5313	0.5353	0.5640	0.5318
Non-hybrid architectures								
LSTM LSTM LSTM LSTM	**0.4568**	**0.4225**	**0.4307**	**0.4321**	**0.4818**	**0.5375**	**0.5714**	**0.4761**
GRU GRU GRU GRU	0.5043	0.4590	0.4907	0.5097	0.5406	0.6141	0.6721	0.5415

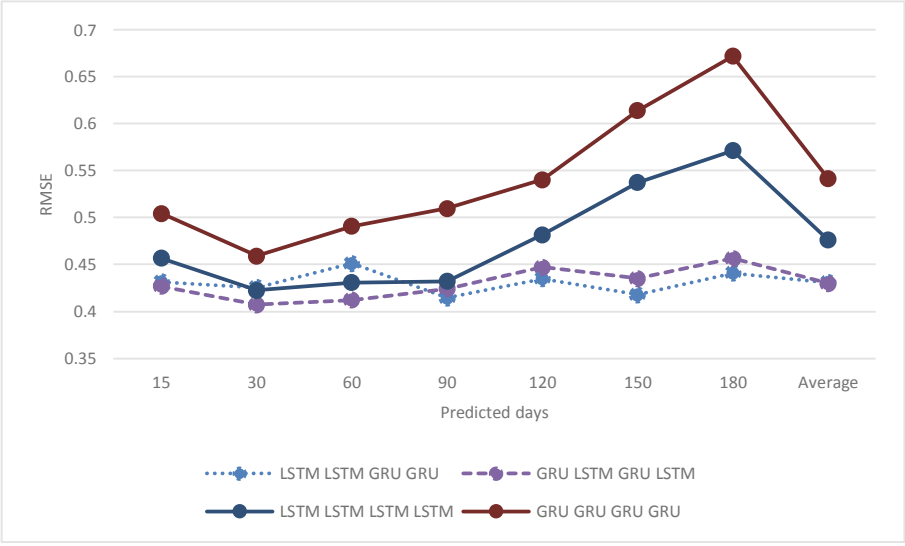


Fig. 8. Comparison between the best hybrid architectures and the non-hybrid.

Figure 8, similar to Fig. 7, shows a lower variability of the RMSEs of hybrid architectures compared to non-hybrid architectures, thus demonstrating its better effectiveness in the prediction task.

According to Table 4, for all sky insolation incident on a horizontal surface with two-layer models on average the hybrid architecture LSTM GRU (RMSE 3.0279) outperformed the non-hybrid architectures. Likewise, it is appreciated that, of the 7 cases of prediction of the time series, in all of them the hybrid LSTM GRU model surpasses the non-hybrid models.

Table 4. Two-layers results for all sky insolation incident on a horizontal surface.

Model	Predicted days							Average
	15	30	60	90	120	150	180	
LSTM GRU	**1.5760**	**1.5464**	**3.8935**	**3.6683**	**3.7493**	**3.4282**	**3.3338**	**3.0279**
GRU LSTM	2.5195	2.0451	4.7721	4.0971	4.2327	3.8511	3.7752	3.6132
Non-hybrid architectures								
LSTM LSTM	**1.6270**	1.9600	4.5927	4.3173	4.3926	4.0273	3.8883	3.5436
GRU GRU	1.9020	**1.6714**	**4.3897**	**3.9588**	**4.0990**	**3.7578**	**3.6914**	**3.3528**

Figure 9 shows a graphical comparison between the hybrid architectures and the non-hybrid.

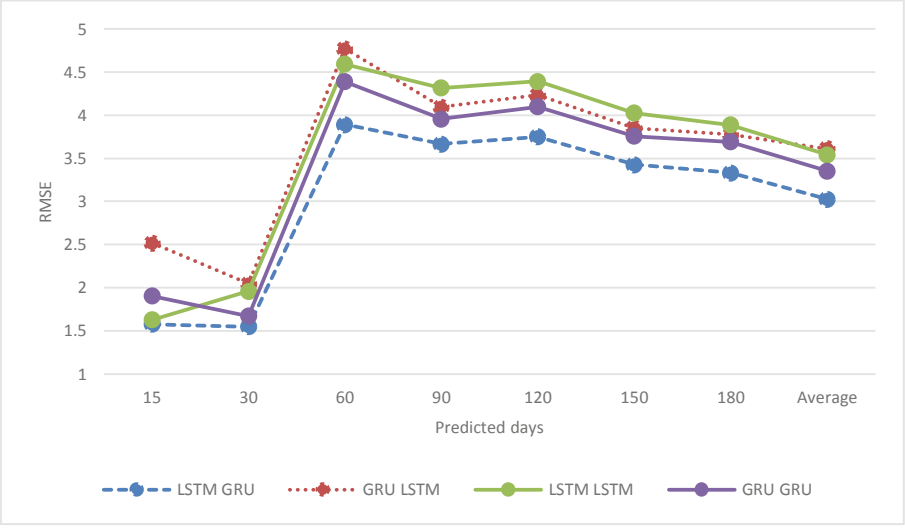


Fig. 9. Comparison between hybrid architectures and the non-hybrid.

In this time series, unlike the previous one, it is appreciated that hybrid and non-hybrid models have similar variability in their RMSEs.

According to Table 5, for all sky insolation incident on a horizontal surface with four-layer models on average the non-hybrid architectures outperformed the hybrid ones, being the best LSTM LSTM LSTM LSTM (RMSE 2.5953). Here it is important to highlight that it is the only case in which none of the proposed hybrid architectures managed to overcome any of the non-hybrid models. Figure 10, shows a graphical comparison between the two best hybrid architectures and the non-hybrids.

Table 5. Four-layers results for all sky insolation incident on a horizontal surface.

Model	15	30	60	90	120	150	180	Average
LSTM LSTM GRU GRU	3.3609	3.9263	6.0509	6.1983	6.2816	5.9595	5.8507	5.3754
GRU GRU LSTM LSTM	1.7583	2.3174	5.5325	5.3325	5.5490	5.2295	5.2260	4.4207
LSTM GRU LSTM GRU	**1.5180**	**1.8676**	**4.2194**	**4.3025**	**4.3880**	**4.0956**	**4.0323**	**3.4890**
GRU LSTM GRU LSTM	3.2353	3.9226	6.8127	6.8240	7.0579	6.7756	6.8005	5.9183
LSTM GRU GRU LSTM	2.8995	3.4487	5.6140	5.7787	5.8555	5.5534	5.4818	4.9473
GRU LSTM LSTM GRU	3.7407	4.3944	6.9911	7.0074	7.1882	6.8734	6.8124	6.1439
Non-hybrid architectures								
LSTM LSTM LSTM LSTM	**1.4821**	**1.3885**	**2.9905**	**3.2183**	**3.2216**	**2.9803**	**2.8862**	**2.5953**
GRU GRU GRU GRU	1.9428	1.6308	3.9893	3.6428	3.7453	3.4272	3.3732	3.1073

Regarding the variability of the RMSEs for predictions of all sky insolation incident on a horizontal surface time series, it is appreciated that in the two-layer models and in the 4-layer models the variability is very similar for hybrid and non-hybrid models.

At this point, it is important to highlight that not all hybrid models produce results that exceed the results of non-hybrid models, therefore, according to the characteristics of the time series to be analyzed, the best non-hybrid models proposed in this work can be or experiment with all those proposed, as well as unbalanced hybrid models can be implemented, for example GRU GRU GRU LSTM, LSTM GRU GRU GRU, etc.

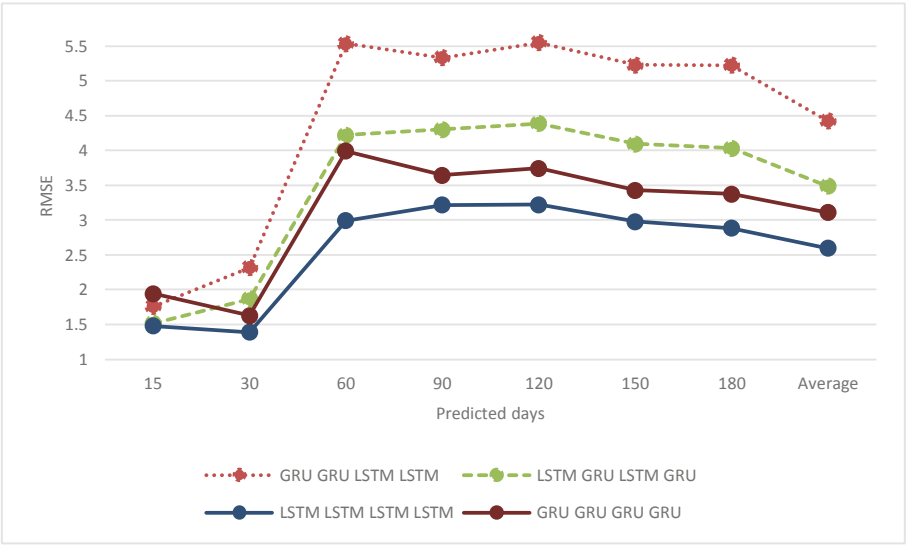


Fig. 10. Comparison between the best hybrid architectures and the non-hybrid.

6 Conclusion

For two-layer architectures, of 14 case studies in 13 of them hybrid architectures outperformed non-hybrid architectures. For 4-layer architectures of 14 case studies, hybrid architectures outperform non-hybrid architectures in 50% of cases. Generalizing from 28 case studies in 21 of them, hybrid architectures surpass non-hybrid ones. Therefore, it is concluded that for the time series analyzed in this study, hybrid architectures based on recurrent neural networks are a great alternative for univariate time series forecasting.

7 Future Work

For future work it would be important to perform the analysis of the proposed hybrid models in time series with different characteristics to those analyzed in this work, for example no-trend and non-seasonality time series.

In addition, it would be interesting to experiment with other hybrid configurations, where the number of layers is not necessarily balanced for each type of recurrent neural network.

References

1. Flores, A., Tito, H., Silva, C.: Local average of nearest neighbors: univariate time series imputation. Int. J. Adv. Comput. Sci Appl. (IJACSA) **10**(8), 45–50 (2019)
2. Flores, A., Tito, H., Centty, D.: Improving long short-term memory predictions with local average of nearest neighbors. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **10**(11), 392–397 (2019)

3. Kyunghyun, C., Bart, V., Caglar, G., Dzmitry, B., Fethi, B., Holger, S., Yoshua, B.: Learning phrase representations using RNN encoder-decoder for statistical machine traslation. arxiv.org, pp. 1–15 (2014)
4. Gail, W., Yoav, G., Eran, Y.: On the practical computational power of finite precision RNNs for language recognition. arxiv.org, pp. 1–9 (2018)
5. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: 30th International Conference on Machine Learning, Atlanta, Georgia, USA (2013)
6. Paco, M., López Del Alamo, C., Alfonte, R.: Forecasting of meteorological weather time series through a feature vector based on correlation. In: 18th International Conference Computer Analysis of Images and Patterns CAIP 2019, Salerno, Italy (2019)
7. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. **35**(8), 1798–1828 (2013)
8. Schmidhuber, J.: Deep learning in neural networks: an overview, arxiv.org (2014)
9. Asiful, M., Karim, R., Thulasiram, R.: Hybrid deep learning model for stock price prediction. In: IEEE Symposium Series on Computational Intelligence, SSCI, Bangalore, India (2018)
10. Fu, Y., Tang, M., Yu, R., Liu, B.: Multi-step ahead wind power forecasting based on recurrent neural networks, In: IEEE PES Asia-Pacific Power and Energy Engineering Conference, APPEEC, Kota Kinabalu, Malaysia (2018)
11. Liu, H., Mi, X., Li, Y., Duan, Z., Xu, Y.: Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional gated recurrent unit network and support vector regression. Renvew. Energy **143**, 842–854 (2019)
12. Titos, M., Bueno, A., García, L., Benitez, M., Ibañez, J.: Detection and classification of continuous volcano-seismic signals with recurrent neural networks. IEEE Trans. Geosci. Remote Sens. **57**(4), 1936–1948 (2018)
13. Elsayed, N., Maida, A., Bayoumi, M.: Deep gated recurrent and convolutional network hybrid model for univariate time series classification. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **10**(5), 654–664 (2019)
14. Diagne, M., David, M., Lauret, P., Boland, J., Schmutz, N.: Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. Renew. Sustain. Energy Rev. **25**, 65–76 (2013)
15. Fouilloy, A., Voyant, C., Notton, G., Motte, F., Paoli, C., Nivet, M., Guillot, E., Duchaud, J.: Solar irradiation prediction with machine learning: forecasting models selection method depending on weather variability. Energy **165**, 620–629 (2018)
16. Lauret, P., Voyant, C., Soubdhan, T., David, M., Poggi, P.: A benchmarking of machine learning techniques for solar radiation forecasting in an insular context. Sol. Energy **112**, 446–457 (2015)
17. Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., Stork, J.: Comparison of different methods for univariate time series imputation in R, arxiv.org (2015)

Evolving Recurrent Neural Networks for Pattern Classification

Gonzalo Nápoles[1,2(✉)]

[1] Faculty of Business Economics, Hasselt University, Hasselt, Belgium
gonzalo.napoles@uhasselt.be
[2] Department of Cognitive Science and Artificial Intelligence, Tilburg University, Tilburg, The Netherlands

Abstract. While reaching outstanding prediction rates by means of black-box classifiers is relatively easy nowadays, reaching a proper trade-off between accuracy and interpretability might become a challenge. The most popular approaches reported in the literature to overcome this problem use post-hoc procedures to explain what the classifiers have learned. Less research is devoted to building classification models able to intrinsically explain their decision process. This paper presents a recurrent neural network—termed Evolving Long-term Cognitive Network—for pattern classification, which can be deemed interpretable to some extent. Moreover, a backpropagation learning algorithm to adjust the parameters attached to the model is presented. Numerical simulations using 35 datasets show that the proposed network performs well when compared with traditional black-box classifiers.

Keywords: Recurrent neural networks · Backpropagation · Evolving Long-term Cognitive Network · Interpretability

1 Introduction

Pattern classification [4] is one of the most popular machine learning problems. Roughly speaking, it consists in assigning the proper category (often referred to as decision class) to each observation. Therefore, state-of-the-art classifiers aim at producing the most accurate prediction rates possible. The rise of *deep learning* [8] has shaken our perception of what is possible within the Artificial Intelligence field. In that regard, pattern classification problems concerned with computer vision have benefited the most. Convolutional neural networks [3,11], deep Bayesian networks [6], deep feed-forward neural networks [2,7], and deep Boltzmann machines [13,19] are representatives of these models.

However, the new EU regulations [9] introducing the right to an explanation for outputs of machine learning models has brought another variable to the equation: the interpretability. This implies that producing outstanding predictions is no longer the only goal to be reached. Reasoning models should be

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 388–398, 2021.
https://doi.org/10.1007/978-3-030-55180-3_29

capable of explaining (to some extent) how they arrived at a particular conclusion. Unfortunately, most accurate classifiers are not well-suited in doing that as they regularly perform as black boxes.

We have two approaches to deal with the EU regulations: either we use post-hoc procedures to elucidate the decision process of existing black-box classifiers, or we develop new interpretable algorithms able to *intrinsically* explain their predictions. An issue with the first approach is that the intuition behind some explanation methods is probably more difficult to understand than the classifier itself, so they are not transparent. On the other hand, these methods often provide just local explanations for a single instance [14], thus proving a limited picture of the problem domain. An issue with the second approach is that reaching a trade-off between interpretability and accuracy can become a nightmare. However, the idea of having a prediction model able to explain its decisions without the need of using a post-hoc method is attractive.

This paper proposes an interpretable neural system, termed *Evolving Long-term Cognitive Network* (ELTCN), to deal with pattern classification problems. The new recurrent model is built upon the Long-term Cognitive Networks (LTCNs)[17] which are deemed interpretable to some extent since all neural concepts in the network have a specific meaning. The distinctive characteristic of this model is that it allows the weights to change from an iteration to another during the reasoning process. While having well-defined neurons help understand the model and its decision process, it posses the problem of obtaining accurate results without having *explicit* hidden neurons. The second contribution of this paper is related to a new backpropagation algorithm which is used to adjust the weights and some transfer function parameters.

The rest of this paper is organized as follows. Section 2 describes the LTCN model. Section 3 presents the proposed neural system, while Sect. 4 elaborates on the backpropagation algorithm. Section 5 contains an empirical study of the developed algorithms. Section 6 concludes the paper.

2 Long-Term Cognitive Networks

Roughly speaking, LTCNs are interpretable recurrent neural networks where each neural concept C_i denotes either an input or output (continuous) variable in a given domain, while the weight $w_{ji} \in \mathbb{R}$ denotes the rate of change in the conditional mean of C_i with respect to C_j, assuming that the activation values of the remaining neurons impacting C_i are fixed. This is similar to interpreting the coefficients in a regression model. Hidden neurons are not allowed as they cannot be interpreted naturally [18]. It also holds that:

- $w_{ji} > 0$: If positively activated, C_j will impact positively C_i. More explicitly, the higher (lower) the activation value of C_j in the current iteration, the higher (lower) the value of C_i in the following iteration;
- $w_{ji} < 0$: If negatively activated, C_j will impact negatively C_i. More explicitly, the higher (lower) the activation value of C_j in the current iteration, the lower (higher) the value of C_i in the following iteration.

Equation (1) shows the reasoning rule of LTCNs, which computes the activation value $A_i^{(t+1)}$ of the neural concept C_i in the $(t+1)$-th iteration for a given input pattern as the initial activation vector,

$$A_i^{(t+1)} = f_i^{(t+1)} \left(\sum_{j=1}^{M} w_{ji} A_j^{(t)} \right) \tag{1}$$

where M denotes the number of neural concepts in the network, whereas $f_i^{(t+1)}$ is the transfer function adopted to confine the activation value of each neuron to the desired interval. The nonsynaptic learning of LTCNs [15,17] is devoted to computing the shape of the sigmoid transfer function $f_i(\cdot)$ in each iteration while preserving the weights defined by human experts.

3 Evolving Long-Term Cognitive Networks

This section presents a new recurrent neural system termed *Evolving Long-term Cognitive Networks* (ELTCNs) for pattern classification.

3.1 Network Architecture

Let $\mathcal{X} = \{X_1, X_2, \ldots, X_M\}$ be a set of numerical variables describing a classification problem such that each problem instance is associated with a decision class. The classification problem [4] consists in building a mapping $\psi : \mathcal{X}^M \rightarrow \mathcal{Y}$ that assigns to each instance the proper decision class from the N possible ones in $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_N\}$. This problem can be modeled by using an LTCN-based architecture. Remark that the added value would probably rely on the model's interpretability rather than its outstanding accuracy.

In the proposed model, problem variables are denoted as input neurons, which can be either dependent or totally independent [16]. The former ones do admit other input neurons to impact them, whereas the latter ones just propagate their initial activation vector and they are not influenced by any other input neuron, therefore their activation values remain static. Output neurons are used to compute the decision class for an initial activation vector.

In the proposed architecture, input neurons are connected with each other (unless the domain experts state otherwise) thus providing the system with a recurrent network topology. Moreover, each input neuron is connected with the output ones. Figure 1 shows, as an example, the topology of an LTCN-based classifier involving three features and three decision classes.

3.2 Evolving Reasoning Rule

One might think that this type of comprehensible neural architecture is "limited" when it comes to the number of hidden neurons and layers. However, any recurrent neural network can be unfolded into a multilayered network with a

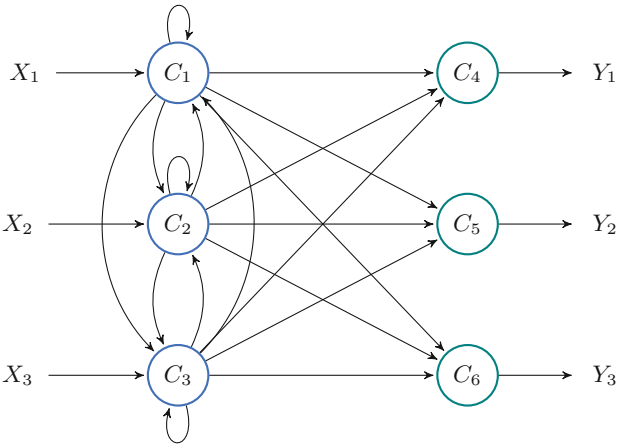


Fig. 1. Neural model comprised of three input neurons (C_1, C_2, C_3) and three output neurons (C_4, C_5, C_6) encoding the decision classes.

fixed width but unlimited length, theoretically speaking. The problem with this multilayered network is that we will have the same weight connecting the neurons C_i and C_j in all of these abstract hidden layers.

The most distinctive feature of the ELTCN model is that it allows the weights to change from an iteration to another. This causes the emergence of T *abstract hidden layers*, each containing M *abstract hidden neurons*, therefore equipping the network with improved learning capabilities.

Equation (2) shows how to compute the activation values of input neurons by following the evolving principle,

$$A_i^{(t+1)} = f_i^{(t+1)} \left(\sum_{j=1}^{M} w_{ji}^{(t)} A_j^{(t)} \right) \tag{2}$$

where $f(\cdot)$ can be either the sigmoid function,

$$s_i^{(t)}(x) = \frac{1}{1 + e^{-\lambda_i^{(t)}(x - h_i^{(t)})}} \tag{3}$$

or the hyperbolic tangent function,

$$q_i^{(t)}(x) = \frac{e^{2\lambda_i^{(t)}(x - h_i^{(t)})} - 1}{e^{2\lambda_i^{(t)}(x - h_i^{(t)})} + 1} \tag{4}$$

such that $\lambda_i^{(t)} > 0$ and $h_i^{(t)} \in \mathbb{R}$ are two parameters denoting the function slope and its offset, respectively. Given N decision classes, the activation values for output neurons will be computed as follows:

$$A_i^{(t+1)} = \frac{e^{\left(\sum_{j=1}^{M} w_{ji}^{(t)} A_j^{(t)}\right)}}{e^{\left(\sum_{k=1}^{N} \left(\sum_{j=1}^{M} w_{jk}^{(t)} A_j^{(t)}\right)\right)}}. \tag{5}$$

4 Backpropagation Learning

This backpropagation algorithm is devoted to estimating the weights and the offset parameter associated with each neural entity. The first step is to perform the forward pass so that we can compute the total error $\mathcal{E}$, which is calculated by using the cross-entropy between the expected response (binary) vector Y and the predicted one for a given instance.

Case 1. When $t = T$, we have the following:

$$\mathcal{E} = -\sum_{i=1}^{N} Y_i log A_i^{(t)} \tag{6}$$

where Y_i is the actual value of the i-th decision class, while $A_i^{(t)}$ is the activation value of that decision neuron. Consequently,

$$\frac{\partial \mathcal{E}}{\partial A_i^{(t)}} = -\frac{Y_i}{A_i^{(t)}} + \frac{1 - Y_i}{1 - A_i^{(t)}}. \tag{7}$$

Since the output neurons use a *softmax* transfer function, we need to calculate the partial derivatives of the output they produce $A_i^{(t)}$ with respect to the raw activation values $\bar{A}_j^{(t)}$. More generically we have:

$$\frac{\partial A_i^{(t)}}{\bar{A}_j^{(t)}} = \begin{cases} \bar{A}_i^{(t)}(1 - \bar{A}_i^{(t)}) & i = j \\ -\bar{A}_i^{(t)} \bar{A}_j^{(t)} & i \neq j \end{cases}. \tag{8}$$

Case 2. When $1 < t < T$, we have the following:

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial A_i^{(t)}} &= \sum_{j=1}^{M} \frac{\partial \mathcal{E}}{\partial A_j^{(t+1)}} \times \frac{\partial A_j^{(t+1)}}{\partial A_i^{(t)}} \\ &= \sum_{j=1}^{M} \frac{\partial \mathcal{E}}{\partial A_j^{(t+1)}} \times \frac{\partial A_j^{(t+1)}}{\partial \bar{A}_j^{(t+1)}} \times \frac{\partial \bar{A}_j^{(t+1)}}{\partial A_i^{(t)}} \\ &= \sum_{j=1}^{M} \frac{\partial \mathcal{E}}{\partial A_j^{(t+1)}} \times \frac{\partial A_j^{(t+1)}}{\partial \bar{A}_j^{(t+1)}} \times w_{ij} \end{aligned} \tag{9}$$

where $\frac{\partial A_j^{(t+1)}}{\partial \bar{A}_j^{(t+1)}}$ is given by $f_j^{'(t+1)}$. For the sigmoid function we have:

$$\frac{\partial A_j^{(t+1)}}{\partial \bar{A}_j^{(t+1)}} = \frac{1}{4} \lambda_j^{(t+1)} \text{sech}^2 \left(\frac{1}{2} \lambda_j^{(t+1)} \left(\bar{A}_j^{(t+1)} - h_j^{(t+1)} \right) \right)$$

while for the hyperbolic tangent we have:

$$\frac{\partial A_j^{(t+1)}}{\partial \bar{A}_j^{(t+1)}} = \lambda_j^{(t+1)} \text{sech}^2 \left(\lambda_j^{(t+1)} \left(\bar{A}_j^{(t+1)} - h_j^{(t+1)} \right) \right).$$

Subsequently, we need to obtain the partial derivatives of the total error with respect to the transfer function parameters $h_i^{(t)}$ as follows:

$$\frac{\partial \mathcal{E}}{\partial h_i^{(t)}} = \frac{\partial \mathcal{E}}{\partial A_i^{(t)}} \times \frac{\partial A_i^{(t)}}{\partial h_i^{(t)}}. \tag{10}$$

– For the sigmoid transfer function we have:

$$\frac{\partial A_i^{(t)}}{\partial h_i^{(t)}} = -\frac{1}{4}\lambda_i^{(t)} \text{sech}^2\left(\frac{1}{2}\lambda_i^{(t)}\left(\bar{A}_i^{(t)} - h_i^{(t)}\right)\right). \tag{11}$$

– For the hyperbolic tangent function we have:

$$\frac{\partial A_i^{(t)}}{\partial h_i^{(t)}} = \lambda_i^{(t)}\left(-\text{sech}^2\left(\lambda_i^{(t)}\left(\bar{A}_i^{(t)} - h_i^{(t)}\right)\right)\right). \tag{12}$$

Finally, Eq. (13) shows the partial derivative of the total error with respect to the weights in the t-th abstract layer,

$$\frac{\partial \mathcal{E}}{\partial w_{ij}^{(t)}} = \frac{\partial \mathcal{E}}{\partial A_j^{(t)}} \times \frac{\partial A_j^{(t)}}{\partial \bar{A}_j^{(t)}} \times \frac{\partial \bar{A}_j^{(t)}}{\partial w_{ij}^{(t)}} \tag{13}$$

such that

$$\frac{\partial \bar{A}_j^{(t)}}{\partial w_{ij}^{(t)}} = \frac{\partial\left(\sum_{l=1}^{M} A_i^{(t-1)} w_{lj}^{(t)}\right)}{\partial w_{ij}^{(t)}} = A_i^{(t-1)}. \tag{14}$$

Equations (15) shows the gradient vector for the transfer function parameters attached to the (input) neurons in the t-th abstract layer. Similarly, Eq. (16) shows the gradient vector corresponding with the weights connecting the t-th abstract layer with the following one.

$$\nabla_h^{(t)}\mathcal{E} = \left(\frac{\partial \mathcal{E}}{\partial h_1^{(t)}}, \ldots, \frac{\partial \mathcal{E}}{\partial h_i^{(t)}}, \ldots, \frac{\partial \mathcal{E}}{\partial h_M^{(t)}}\right) \tag{15}$$

$$\nabla_w^{(t)}\mathcal{E} = \left(\frac{\partial \mathcal{E}}{\partial w_{11}^{(t)}}, \ldots, \frac{\partial \mathcal{E}}{\partial w_{ij}^{(t)}}, \ldots, \frac{\partial \mathcal{E}}{\partial w_{MM}^{(t)}}\right). \tag{16}$$

The reader can notice that such gradient vectors are all we need to adjust the network parameters by means of a gradient descent method. In this paper, the Adam optimization method [10] was used to perform the numerical simulations. It is worth mentioning that the proposed backpropagation algorithm could be adjusted to deal with a large number of abstract layers coming from the reasoning process, however, that is beyond the scope of this paper.

5 Numerical Simulations

This section is devoted to evaluating the discriminatory power of the proposed ELTCN model on 35 pattern classification datasets. Such benchmark problems (see Table 1) have been collected from both the KEEL [1] and UCI ML [12] repositories. A brief inspection of these datasets shows that the number of instances goes from 106 to 10,9992. The number of attributes ranges from 4 to 29 while the number of decision classes goes from 2 to 15.

The state-of-the-art classifiers selected for comparison include: Support Vector Machines (SVM), Gaussian Naive Bayes (GNB), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Multilayer Perceptron (MLP) using two hidden layers, each with 50 ReLU neurons. Moreover, we used the ADAM optimization algorithm and the sparse cross-entropy to measure the error during the training process. The numbers of epochs in ADAM was set to 200 in all cases to keep the simulation time low. The parameters attached to the remaining models were configured as provided in *Scikit-learn*. Hence, no model performed hyperparameter tuning, which is convenient to assess their performance when their parameters have not been optimized for each dataset.

This experiment includes two ELTCN models, both performing 5 iterations during the recurrent inference process. The first model (ELTCN-S) uses sigmoid transfer functions, while the second one (ELTCN-H) uses hyperbolic tangent ones. Figure 2 shows the average prediction rates on the 35 datasets, after performing 10-fold cross-validation. It can be noticed that RF reports the highest prediction rates, followed by ELTCN-H and MLP, while the ELTCN-S variant ranked last. The poor performance of ELTCN-S could be a result of the saturation issues of the sigmoid transfer function.

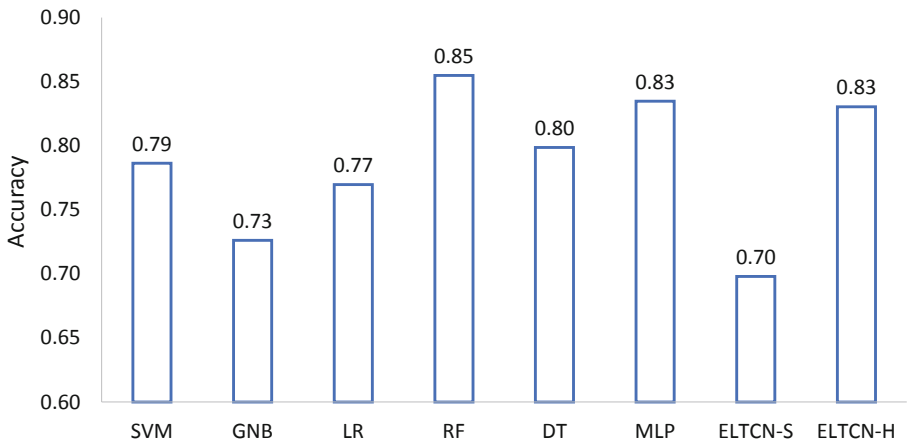


Fig. 2. Average prediction rate achieved by each classifier on the 35 benchmark problems adopted for simulation purposes.

Table 1. Benchmark problems used during the simulations.

ID	Name	Instances	Attributes	Classes	Noisy
DS1	Acute-inflammation	120	6	2	No
DS2	Acute-nephritis	120	6	2	No
DS3	Appendicitis	106	7	2	No
DS4	Cardiotocography-10	2,126	20	10	No
DS5	Collins	500	22	15	No
DS6	Dermatology	366	16	6	No
DS7	Echocardiogram	131	11	2	No
DS8	Ecoli	336	7	8	No
DS9	Flags	194	29	8	No
DS10	Glass	214	9	6	No
DS11	Heart-5an-nn	270	13	2	Yes
DS12	Heart-statlog	270	13	2	No
DS13	Iris	150	4	3	No
DS14	Libras	360	27	15	No
DS15	Liver-disorders	345	6	2	No
DS16	Mfeat-morphological	2,000	7	10	No
DS17	Mfeat-zernike	2,000	26	10	No
DS18	Monk-2	432	6	2	No
DS19	New-thyroid	215	5	2	No
DS20	Page-blocks	5,473	11	5	No
DS21	Parkinsons	195	22	2	No
DS22	Pendigits	10,992	14	10	No
DS23	Pima	768	8	2	No
DS24	Planning-relax	182	12	2	No
DS25	Saheart	462	9	2	No
DS26	Segment	2,310	19	7	No
DS27	Vehicle	846	18	4	No
DS28	Vertebral2	310	6	2	No
DS29	Vertebral3	310	6	3	No
DS30	Wall-following	5,456	25	4	No
DS31	Wine	178	13	3	No
DS32	Wine-5an-nn	178	13	3	Yes
DS33	Winequality-red	1,599	11	6	No
DS34	Winequality-white	4,898	11	7	No
DS35	Yeast	1,484	8	10	No

To determine whether the observed performance differences are statistically significant or not, the Friedman test [5] has been conducted. This non-parametric test will reject the null hypothesis when at least two classifiers perform significantly different. The p-value=0.0 < 0.05 suggests rejecting the null hypothesis for a 95% confidence interval. However, this does not mean that we can certainly conclude that the ELTCN-H method (best-performing variant) significantly contributes to the differences reported by the Friedman test.

Therefore, the following analysis is dedicated to performing pairwise comparisons with ELTCN-H being the control algorithm. Such pairwise comparisons are conducted using the Wilcoxon signed-rank test [20]. Table 2 shows the p-values reported by this non-parametric test, the negative (R^-) and the positive (R^+) ranks, the corrected p-values according to Holm, and whether the null hypothesis was rejected or not for a significance level of 0.05.

Table 2. Wilcoxon pairwise analysis with Holm correction for a significance level of 0.05 such that ELTCN-H is used as the control classifier.

Algorithm	p-value	R^-	R^+	Holm	Hypothesis
SVM	9.345E-03	11	21	2.804E-02	Reject
GNB	2.476E-06	4	30	1.733E-05	Reject
LR	4.710E-04	8	24	2.355E-03	Reject
RF	1.942E-02	23	9	3.884E-02	Reject
DT	4.890E-03	9	24	1.956E-02	Reject
MLP	2.452E-01	17	13	2.452E-01	Fail to reject
ELTCN-S	3.247E-06	3	30	1.948E-05	Reject

The reader can observe that the ELTCN-H variant outperformed all classifiers but RF and MLP. What is perhaps more important is that ELTCNs have a shallow architecture in which each neuron has a specific meaning for the domain problem. This makes ELTCNs quite interpretable, as opposed to traditional neural systems that operate as black boxes.

6 Conclusions

The results suggest that the ELTCN-H model is able to produce competitive prediction rates when compared with traditional classifiers. This feature in conjunction with the interpretability of ELTCNs supports the attractiveness of the proposed neural system. The reader can notice, however, that this research does not elaborate on the interpretability issues. Instead, it is assumed that we could understand how the decision classes were derived by simply inspecting the weights describing the system. This naive assumption oversimplifies the complexity behind an FCM-based classifier. The future work will focus on shedding light on the interpretability of the proposed neural system.

References

1. Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Multiple Valued Logic Soft Comput. **17**(2–3), 255–287 (2011)
2. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: training deep neural networks with binary weights during propagations. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., (eds.) Advances in Neural Information Processing Systems, vol. 28, pp. 3123–3131. Curran Associates, Inc., (2015)
3. Derevyanko, G., Grudinin, S., Bengio, Y., Lamoureux, G.: Deep convolutional networks for quality assessment of protein folds. Bioinformatics **34**(23), 4046–4053 (2018)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classif., 2nd edn. Wiley, Hoboken (2012)
5. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **32**(200), 675–701 (1937)
6. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, vol. 70, pp. 1183–1192. JMLR.org (2017)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
9. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Mag. **38**(3), 50–57 (2017)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates Inc., (2012)
12. Lichman, M.: UCI machine learning repository (2013)
13. Melchior, J., Fischer, A., Wiskott, L.: How to center deep boltzmann machines. J. Mach. Learn. Res. **17**(99), 1–61 (2016)
14. Molnar, C.: Interpretable Machine Learning (2019). https://christophm.github.io/interpretable-ml-book/
15. Nápoles, G., Salmeron, J.L., Vanhoof, K.: Construction and supervised learning of long-term grey cognitive networks. IEEE Transactions on Cybernetics, 1–10 (2019)
16. Nápoles, G., Espinosa, M.L., Grau, I., Vanhoof, K., Bello, R.: Fuzzy cognitive maps based models for pattern classification: advances and challenges. In: Pelta, D., Cruz Corona, C. (eds.) Soft Computing Based Optimization and Decision Models. Studies in Fuzziness and Soft Computing, vol. 360, pp. 83–98. Springer, Cham (2018)
17. Nápoles, G., Vanhoenshoven, F., Falcon, R., Vanhoof, K.: Nonsynaptic error backpropagation in long-term cognitive networks. IEEE Trans. Neural Netw. Learn. Syst. **31**(3), 865–875 (2019)
18. Nápoles, G., Vanhoenshoven, F., Vanhoof, K.: Short-term cognitive networks, flexible reasoning and nonsynaptic learning. Neural Netw. **115**, 72–81 (2019)

19. Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: van Dyk, D., Welling, M., (eds.) Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 5, pp. 448–455. PMLR, 16–18 April 2009
20. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics **1**, 80–83 (1945)

Neural Network Modeling of Productive Intellectual Activity in Older Adolescents

Sipovskaya Yana Ivanovna(✉)

Institute of Psychology of the Russian Academy of Sciences, Moscow State Psychological and Pedagogical University, Yaroslavskaya st., 13, 129366 Moscow, Russia
syai@mail.ru

Abstract. The magnitude of the tasks facing modern society increases the relevance of the problem of human competence, e.g. intellectual productivity (in professional and everyday activities). However, the problem of intellectual competence is open to research. The purpose of this study: to determine the construct of "intellectual competence" using neural network programming. Intellectual competence is a special type of organization of knowledge that provides the ability to make effective decisions in a certain subject area, so we can talk about the competence of a poker player, a schoolboy, etc. Thus, we have identified as the basis of intellectual competence several components: conceptual, metacognitive and intentional abilities, which manage of human mental activity; Study participants: 90 students at the age of 15 years; and Techniques: "Conceptual Synthesis", "Method for Diagnosing the Degree of Development of Reflexivity", "Comparison of Similar Figures", "Intentions" and "Interpretation". In accordance with the obtained results, it can be concluded that a multi-level construct of intellectual competence in older adolescents can be modeled using neural network analysis with an accuracy of the constructed model of 99.5%. The residual variance explained by the complexity and heterogeneity of the structure of intellectual competence and the participants' transitional age.

Keywords: Productive intellectual activity · Neural network modeling · Intellectual competence · Abilities

1 Introduction

Features of adolescence determine the dramatic and significant changes in all aspects of human life. In the context of intellectual development, this period is characterized by an increase in mental capabilities due to the maturation of conceptual, metacognitive and intentional abilities that are useful for the implementation of productive intellectual activity in a certain field, that is, intellectual competence.

It should be noted that the specificity and properties of competence, including intellectual competence, are manifested almost exclusively in any specific real practical activity. In this regard, new questions arise regarding nature and the essence of the competence construct. Is there a component of competence in any activity that is not subject

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 399–406, 2021.
https://doi.org/10.1007/978-3-030-55180-3_30

to consciousness, or is it inherent only to a certain age, is it like a transitional stage to a more advanced way of thinking?

Regardless of how theoretical problems are solved, the study of competence, its features and properties is of great interest and will undoubtedly be useful both in the light of a deeper understanding of the thinking mechanisms themselves and possible vectors of improvement, development of practical activities and optimization of expended resources.

This article attempts to determine intellectual competence and its structural components using neural network modeling, as well as the results of an empirical study that reveals the relationship of conceptual, metacognitive and intentional experience in the framework of the construct of intellectual competence.

In a number of studies, the components of intellectual competence were identified, among which cognitive, metacognitive, personality, motivational indicators were identified:

1) subject knowledge [8, 10];
2) conceptual, categorical, semantic abilities [8, 9, 12, 13];
3) intellectual self-regulation [3, 7, 8, 16];
4) intentional knowledge [9, 16];
5) specific motivation [4, 5, 9, 11, 14];
6) the quality of thinking, namely: cognitive need, flexibility, criticality, creativity [13–16].

There were investigated: 1) manifestations of intellectual competence as the ability to interpret (transforming individual experience); 2) voluntary and involuntary metacognitive abilities; 3) intentional abilities; 4) conceptual (generative) abilities in present work.

1.1 Research Questions

The theoretical hypothesis of the study: the ratio of conceptual, metacognitive and intentional experience is an important factor determining the uniqueness of individual intellectual competence.

1.2 Purpose of the Study

Purpose: to reveal the neuronal structure of intellectual competence in its cognitive components.

The objective of this study is to determine the neuronal structure of the construct of intellectual competence in the context of manifestations of conceptual abilities, voluntary and involuntary metacognitive abilities, and intentional abilities.

Thus, the subject of this study is the neuronal network structure of intellectual competence, the object of study is students at the age of 15 years who have intellectual competence formed in the process of learning.

2 Study Participants

The sample consisted of 90 students (54 girls and 36 boys) at the age of 15 years.

3 Research Methods

3.1 Method of Diagnosing Conceptual Abilities. "Conceptual Synthesis" [9]

Indicator: generative (conceptual) abilities.

3.2 Methods for Identifying Metacognitive Abilities

Methods of Diagnosing the Degree of Development of Reflexivity [7]
Indicators: reflexivity.

Comparison of Similar Drawings [6]
Indicators: 1) latent time of the first response (amount); 2) the total number of errors.

3.3 Method of Diagnosis of Intentional Abilities "Intentions" [15]

Indicators: 1) mindset, 2) beliefs.

3.4 Methodology for Assessing Intellectual Competence "Interpretation" [1, 2, 13, 17]

Indicators: intellectual competence (total score).

4 Results

In connection with the previously obtained facts about the descriptors of intellectual competence, we used a neural network to build a regression model of the construct of intellectual competence. In the data, seven variables are represented: six independent, predictors and one - target, dependent. At the first stage of the analysis, we built a three-dimensional scattering diagram in order to see how this data is distributed in space.

The results of this analysis are presented in Fig. 1.

We see that the data are linear in shape and form three compact clusters located along the main discriminants of intellectual competence.

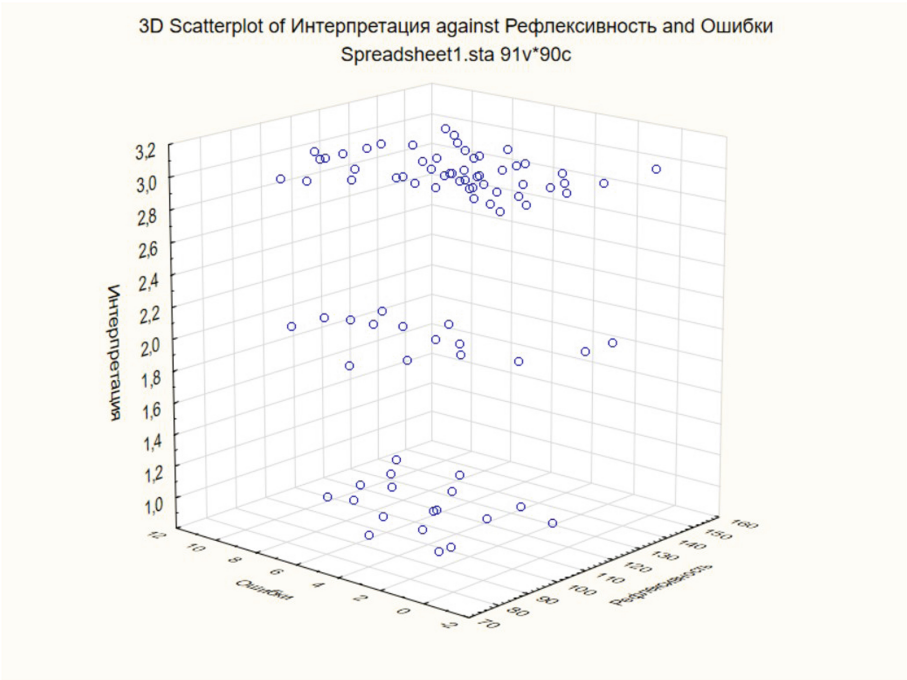


Fig. 1. Network model of intellectual competence

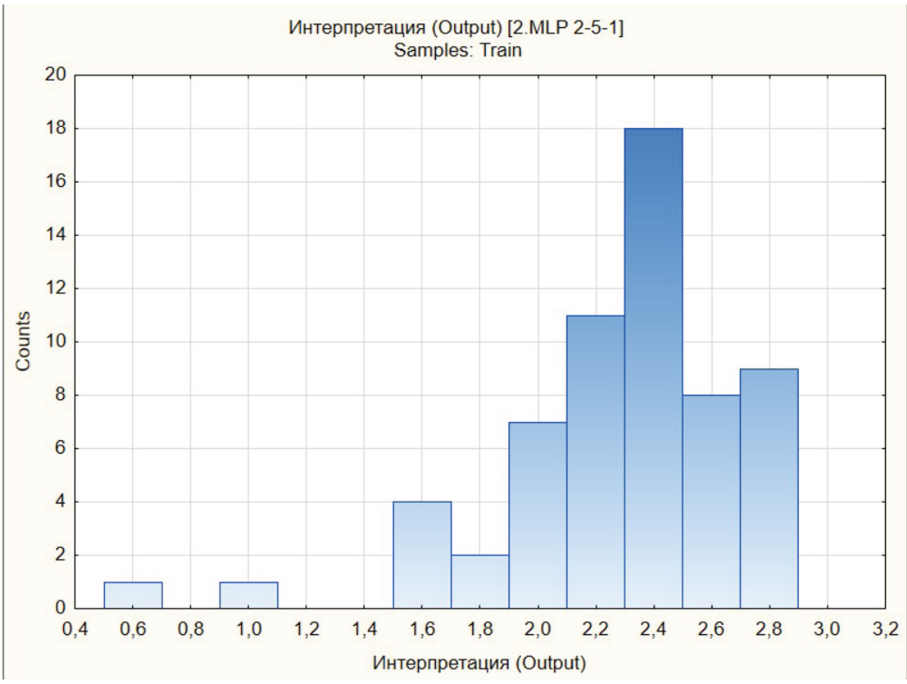
There were no significant emissions beyond the boundaries of the range of interest to us; accordingly, we started building neural network models of intellectual competence.

Due to the fact that we did not have pre-built hypotheses regarding the structure of the resulting neural network, we used the strategy of automated selection of the perceptron type neural network and got five models. Having trained the obtained networks through two training modes: a user neural network and interactive training (multiple subsamples), we received a set of five constructed networks, presented in Table 1.

Table 1. Resulting neural networks of intellectual competence

Index	Net. name	Training performance	Test performance	Validation performance	Training algorithm	Output activation
1	MLP 2-3-1	0,416510	0,592603	0,779625	0,298961	0,214043
2	MLP 2-5-1	0,410779	0,763825	0,778725	0,302728	0,197014
3	MLP 2-8-1	0,445119	0,622301	0,608530	0,289715	0,194303
4	MLP 2-5-1	0,380366	0,717208	0,634722	0,311312	0,198481
5	MLP 2-4-1	0,415067	0,782290	0,687751	0,299550	0,167666

In further analysis, to verify the adequacy of the constructed networks, we constructed a histogram of the most plausible and reliable model presented in Histogram 1.



Hist. 1. Network model 15 MPL 6-6-1 of intellectual competence.

Next, we built a response surface to demonstrate the relationship between the components of intellectual competence and the objective function, which is presented in Fig. 2.

The response surface demonstrated the previously obtained fact that manifestations of high intellectual productivity, i.e. intellectual competence are also characterized by a high degree of formation of conceptual and arbitrary metacognitive abilities.

Choosing for further analysis the 15th model that has the highest performance in the control sample, we checked its quality in the test sample (Table 2).

According to the Table 2, one can make an input that a multi-level construct of intellectual competence in older adolescents can be modeled using neural network analysis with an accuracy of the constructed model of 99.5%. The residual variance explained by the complexity of the structure of intellectual competence and the transitional age of the study participants. The method of neural network modeling of productive intellectual activity revealed the structure if the construct of intellectual competence within conceptual, voluntary metacognitive and intentional abilities, while arbitrary metacognitive abilities are not so important at the neural network structure.

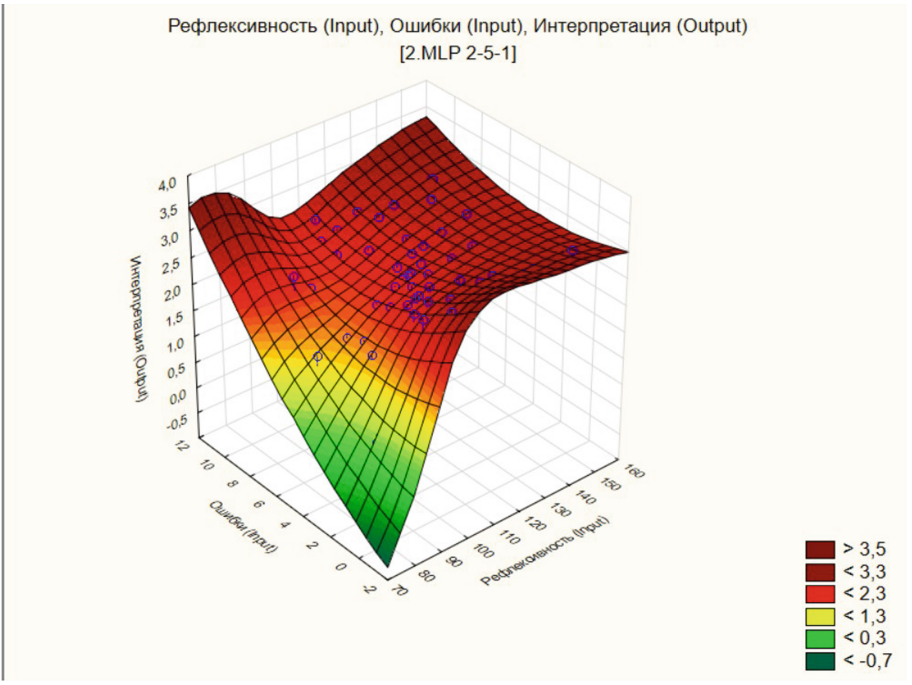


Fig. 2. Response surface.

Table 2. Quality in the test sample of the 15th model

Case name	Interpretation Target	Interpretation-Output 15. MLP 6-6-1.	Interpretation-Abs. Res. 15. MLP 6-6-1.	Mape V3/Abs(V1)	Mean Case 1–13
2	3,0000000	2,892991	0,107009	0,03566972	0,0503109704
9	3,0000000	3,011474	0,011474	0,00382483	
10	3,0000000	2,832345	0,167655	0,05588486	
27	3,0000000	2,868030	0,131970	0,04399011	
28	3,0000000	3,015313	0,015313	0,00510422	
29	3,0000000	2,951831	0,048169	0,01605619	
40	1,0000000	1,190589	0,190589	0,19058901	
49	3,0000000	2,740186	0,259814	0,08660473	
61	3,0000000	2,767422	0,232578	0,07752606	
63	3,0000000	3,020743	0,020743	0,00691437	
68	3,0000000	2,832038	0,167962	0,05598747	
72	3,0000000	2,936764	0,063236	0,02107877	
73	2,0000000	1,890375	0,109625	0,05481225	

5 Findings

The results obtained in this study allowed us to form a neural model that describes the structure of the construct of intellectual competence with its most significant components - conceptual, metacognitive and intentional abilities. The dedicated neural network model is highly accurate and predictive in strength, which will make it possible to use it in any such studies of human intellectual abilities. In addition, a high predictive power allows further use of the obtained neural network for other samples - gerontological and cross-cultural. Thus, we have become one step closer to understanding the underlying mechanisms that underlie individual intellectual resources.

Acknowledgments. The study was carried out by a grant from the Russian Science Foundation (project 19-013-00294).

References

1. Asmolov, A.G.: Formation of Universal Educational Actions in Primary School: From Action to Thought, pp. 57–61. Enlightenment, Moscow (2010)
2. Asmolov, A.G.: Psychology of Personality. Publishing House of Moscow State University, Moscow (1990)
3. Berestneva, O.G.: Modeling the development of intellectual competence of students. Bull. Tomsk Polytech. Univ. **308**(2), 152–156 (2005)
4. Chamorro-Premuzic, T., Furnham, A.: Intellectual competence. Psychologist **18**(6), 352–354 (2005)
5. Chamorro-Premuzic, T., Arteche, A.: Intellectual competence **36**. 564–573 (2008)
6. Kagan, J.: Reflection-impulsivity: the generality and dynamics of conceptual tempo. J. Abnorm. Psychol. **71**, 17–24 (1966)
7. Karpov, A.V.: Reflexivity as a mental property and a method for its diagnosis. Psychol. J. **24**(5), 45–57 (2003)
8. Kholodnaya, M.A., Berestneva, O.G., Kostrikina, I.S.: Cognitive and metacognitive prerequisites of intellectual competence in the field of scientific and technical activities. Psychol. J. **26**(1), 51–59 (2005)
9. Kholodnaya, M.A., Trifonova, A.V., Volkova, N.E., Sipovskaya, Y.I.: Diagnostic techniques for conceptual abilities. Exp. Psychol. **12**(3), 105–118 (2019)
10. Kim, O.G.: Educational activities in small groups as a factor in the optimization of music lessons in elementary school: dis…. Cand. ped. sciences. M (2013)
11. Kiseleva, T.S.: Research of the initiative of the person in intellectual activity. In: Psychology of Individuality: Materials of the All-Russian Conference, 2–3 November 2006, Moscow, pp. 220–223. M (2006)
12. Kozlova, N.V., Sivitskaya, L.A., Katchalov, N.A.: Innovative educational technologies as a condition for the development of professional competencies of higher school teachers. News Tomsk Polytech. Univ. **309**(4), 240–243 (2006)
13. Sipovskaya, Ya.I.: Conceptual, metacognitive and intentional descriptors of intellectual competence in older adolescence. SPSU Bull. **12**(4), 22–31 (2015)
14. Sipovskaya Ya.I.: Descriptors of intellectual competence in older adolescence. Psychology and Pedagogy: Theoretical and Practical Aspects of Modern Sciences. In: Proceedings of the XXVII International Scientific and Practical Conference, pp. 34–36. M. (2014)

15. Sipovskaya Ya.I. The structure of intellectual competence in older adolescence. In: Zhuravleva, A.L., Sergienko, E.A., Kharlamenkova, N.E., Zueva, K.B. (eds.) Psychology - The Science of the Future: Materials of the V International Conference of Young Scientists, p. 272. M. (2013)
16. Sultanova, L.B.: The Problem of Implicit Knowledge in Science. Publishing house UGNTU, Ufa (2004)
17. Yadrovskaya, E.R.: The development of interpretive activity of the student-student in the process of literary education (grades 5–11): dis. … Dr. Ped. sciences. St. Petersburg (2012)

Bidirectional Estimation of Partially Black-Boxed Layers of SOM-Based Convolutional Neural Networks

Ryotaro Kamimura[1,2(✉)]

[1] Kumamoto Drone Technology and Development Foundation, Techno Research Park, Techno Lab 203, 1155-12 Tabaru Shimomashiki-Gun, Kumamoto 861-2202, Japan
[2] IT Education Center, Tokai University, 4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan
ryotarokami@gmail.com

Abstract. The present paper aims to propose a new type of method, where complicated and incompressible components are temporarily black-boxed, to apply model compression for interpretation to more complicated networks. Then, those partially black-boxed components are estimated in forward and backward ways, namely, bi-directional estimation. Finally, the network with the estimated component is compressed as much as possible to have the simplest prototype model for easy interpretation. Then, we try to estimate the black-boxed layers in a new type of network architecture where the self-organizing map (SOM) is combined with the convolutional neural network (CNN) to process two-dimensional feature maps from the SOM. The method was applied to two well-known data sets, namely, the pulsar and occupancy detection data sets. In both experiments, we could successfully improve the generalization. In addition, the final compressed weights were very close to the correlation coefficient between inputs and targets of the original data sets for easy interpretation.

Keywords: Bi-directional · Compression · Partially black-boxed · Interpretation · SOM · CNN · Generalization

1 Introduction

Machine learning has shown brilliant success in many application fields recently. As well, many types of methods have penetrated into our daily life, causing much confusion for ordinary people [29]. This is because one of the main problems of machine learning, and in particular the neural networks dealt with in this paper, have been considered one of the typical black-box models where it is almost impossible to interpret the main inference mechanism. In particular, we have much difficulty in persuading ordinary people to accept the decisions made by machine learning, because it is impossible to explain the main inference in

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 407–427, 2021.
https://doi.org/10.1007/978-3-030-55180-3_31

ordinary language. Thus, it is absolutely necessary to respond to the right-to-explanation requests [9], and it is urgently needed to develop interpretation methods.

In this context, the recent rapid development in convolutional neural networks, particularly applied to image and visual data sets, has been accompanied by many different kinds of methods to interpret the convolutional operations and show how the features can be extracted in many convolutional and higher layers. For example, to visualize the features in the higher layers, the methods of maximizing the activations, sampling, and linear combination of previous layers' filters were used [8]. A masking method for the salient parts of input images was proposed for fast and model-agnostic saliency detection [5]. For visual explanation with a high-resolution and class-discrimination property, the gradient-weighted class activation mapping was proposed [23]. The smooth grad [25] is a method to reduce the visual noises by sampling images by adding some noises to the images and taking their averages. Thus, it seems that the internal mechanism has been gradually clarified, at least, in the field of convolutional neural networks. Though those methods have been popular as interpretation methods for the convolutional neural networks, there have already been some concerns about those methods. For example, the experimental results have already shown that those visualization methods do not satisfy the input invariance, giving misleading interpretation results [16]. In addition, some experimental results have already confirmed that the random initialized weights could produce a visual explanation similar to those by the learned weights [1].

For the internal interpretation of multi-layered neural networks, we propose a novel approach called "neural compressor" to compress multi-layered neural networks into the simplest ones without hidden layers [14] and with important characteristics. First, relations between inputs and outputs should be understood by interpreting all possibilities among the relations. In actual cases, the relations between inputs and the corresponding outputs can be represented by taking into account all the possible routes from the inputs. Second, the interpretation should consider all possible realizations from different input patterns and initial conditions. We think that all possible realizations from different neural networks should have equal importance, and we should take into account all possible ones as much as possible. The interpretation has been so far confined to one individual case, and the compressor has tried to interpret all the possible realizations as much as possible.

Though the neural compressor has shown good applicability for many different data sets, one of the main problems is that there are many cases where the compression operations cannot be applied. For example, when the neighboring layers are connected in too complicated ways, or in extreme cases, when those layers may be disconnected for some reason, the compression cannot be applied naturally. For this problem, we propose a new approach where some specific components to which the compression cannot be applied are black-boxed, and the behaviors of the components are inferred from the inputs and outputs from the parts.

For application, we use the SOM-based convolutional networks. The self-organizing maps (SOM) have been well known as a method to visualize complex input patterns over a two-dimensional feature space [17,18], and many different types of methods have been developed so far [7,15,21,24,26,30–35]. In addition, the SOM knowledge can be naturally used in training supervised learning, and many attempts have been made to extend the SOM to supervised learning [2,6,10–12,17–19,22,27,28]. From our viewpoint, those approaches have not necessarily been successful. One of the main reasons is that the SOM knowledge has been represented mainly over two-dimensional feature spaces, and the conventional neural networks have not the abilities to deal with those two-dimensional feature spaces.

To overcome this problem with the two-dimensional SOM knowledge, we introduce here the convolutional neural networks (CNN), because the CNN have produced many good results for large-scaled visual data sets as well as natural language processing. However, one of the main problems of CNN is that we have had difficulty in interpreting the inference mechanism inside. As explained above, many different types of interpretation methods have been developed, but those approaches, from our viewpoint, have been dependent on the characteristics, namely, image and visual, of data sets. Image and visual data are easily and intuitively understood. When the CNN is applied to the more abstract inputs, used in this paper, it is almost impossible to trace the inference mechanism. Thus, here we temporally black-box the CNN component whose inference mechanism is inferred by examining the inputs and outputs from the CNN components.

Then, we must propose how to estimate the black-boxed components to compress the original multi-layered neural networks. The estimation of black-boxed components is accompanied by information loss in multi-layered neural networks. As is well known, information content in inputs as well as targets gradually disappears when information in many layers is processed step by step. Thus, due to this information loss, we have a serious problem of estimating the black-boxed components. For overcoming this problem of information loss, we introduce here the bi-directional estimation method with forward and backward information augmentation. In the interpretation method, there are several methods to estimate the inner processing in forward and backward ways. For example, one of the well-known methods in the interpretation method is the layer-wise relevance propagation [3] with the relevance conversation property. However, we do not deal here with the relevance but a mechanism to produce the relevance. Thus, it is necessary to deal with the information in inputs and targets in forward and backward ways. At this point, we propose a method to extract information in inputs and targets, necessary to estimate the black-boxed components, and they are processed in forward and backward ways for clearer estimation.

2 Theory and Computational Methods

2.1 Collective Interpretation

We have proposed the collective interpretation for interpreting the internal mechanism of neural networks [13]. We assume that neural networks should be internally interpreted, and in addition, all the realizations from different input patterns and initial conditions should be taken into account. Neural networks have been considered one of the major black-boxed models in machine learning techniques, because of complex network architecture and uncertainty connected with different initial conditions and input patterns. However, we suppose that this uncertainty in producing many different kinds of internal representations should be considered one of the major good points, meaning that a data set can be viewed from a number of different points. In the collective interpretation, we have two operations, namely, horizontal and vertical compression. In the horizontal compression, all the routes from inputs to outputs should be taken into account to represent the collective routes from the input to output. In the vertical compression, all those weights from different initial conditions and input patterns should be taken into account, and actually averaged. Thus, the present method of collective interpretation aims not to interpret one realization of neural networks but to deal with all collections of neural networks for different initial conditions and input patterns (Fig. 1) .

First, let us explain the horizontal compression in more detail. We suppose for simplicity that, for different initial conditions, different weights can be obtained, but in actual situations, for different initial conditions and different input patterns, a number of different weights can be produced. We suppose that, for the tth initial condition ($t = 1, 2, ..., r$), two connection weights of the last two layers, namely, ${}^t w_{jj'}^{(5)}$ and ${}^t w_{ij}^{(6)}$, are combined into

$$ {}^t w_{ij'}^{(6*5)} = \sum_{j=1}^{n_5} {}^t w_{ij}^{(6)} {}^t w_{jj'}^{(5)} \tag{1} $$

Then, supposing that ${}^t w_{ij'}^{(6*3)}$ denotes the compressed weights from the sixth layer to the third layer, the final compressed weights are

$$ {}^t w_{ik}^{(6*2)} = \sum_{j=1}^{n_2} {}^t w_{ij}^{(6*3)} {}^t w_{jk}^{(2)} \tag{2} $$

Then, the vertical compression is the ordinary average operation; in other words, the collective weights can be obtained by averaging all connection weights

$$ \bar{w}_{ik}^{(6*2)} = \frac{1}{r} \sum_{t=1}^{r} {}^t w_{ik}^{(6*2)} \tag{3} $$

The final and compressed weights represent the relations between inputs and outputs directly.

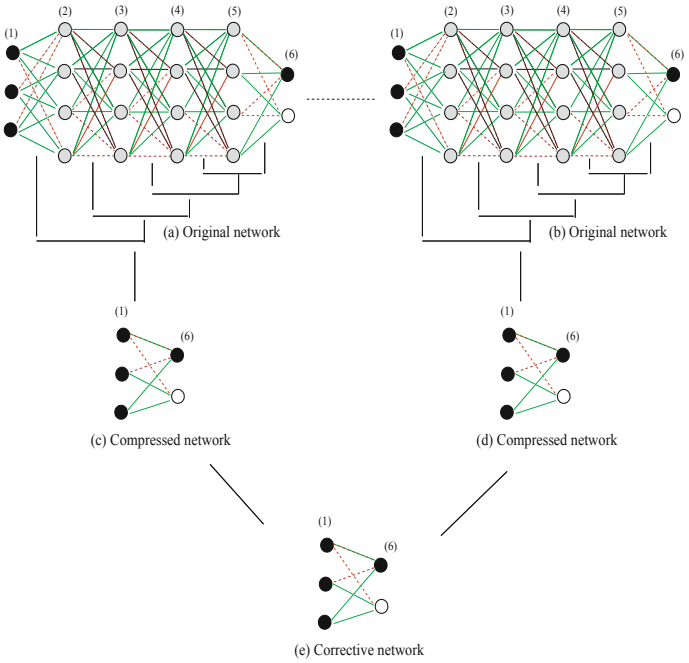


Fig. 1. Network architecture with six layers, including four hidden layers to be compressed immediately into the simplest one.

2.2 Bi-directional Estimation of Black-Boxed Layers

The above conventional compression can be applied to conventional multi-layered neural networks without specific and complex configurations of connection weights. However, for some cases with complex configurations, it is quite difficult to trace all routes from inputs to outputs, because of too complicated and truncated routes. For those cases, we temporarily black-box those complicated components, and we try to estimate the content of complicated components by using the interpretable components.

Let us show an example of the partially black-boxed interpretation by using the six-layered neural networks in Fig. 2(a). We suppose that connection weights between the second and fifth layer cannot be interpreted because of too complicated or truncated problems between them. Thus, we must estimate relations between the second and fifth layer by using the knowledge between the first to the second and the fifth to the sixth layer.

We have two types of estimation procedures, namely, the forward and backward methods. In the forward method, the estimation is based on the normal flow of information from the input to the output. The backward method traces the information route from the output to the input layer. Let us take a network shown in Fig. 2(b), which can be obtained by black-boxing components from the

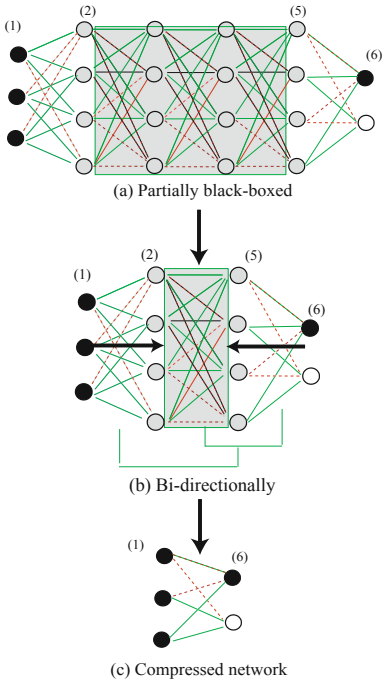


Fig. 2. Network architecture with six layers, including four hidden layers.

second to the sixth layer. First, we should compute the backward output z from the fifth layer

$$ {}^{s}z_{j}^{(5)} = f\left(\sum_{i=1}^{n_6} w_{ij}^{(6)} {}^{s}t_i\right) \tag{4} $$

where ${}^{s}t_i$ is the target for the ith output neuron. Then, the output from the second layer can be computed by

$$ {}^{s}z_{j'}^{(2)} = f\left(\sum_{j=1}^{n_5} w_{jj'}^{(5)} {}^{s}z_{j}^{(5)}\right) \tag{5} $$

The target for the second layer should be the output from the second layer, and it can be computed in a forward way

$$ {}^{s}v_{j'}^{(2)} = f\left(\sum_{k=1}^{n_1} w_{j'k}^{(2)} {}^{s}x_k\right) \tag{6} $$

The final error is computed by

$$ E = \sum_{s=1}^{q} \sum_{j'=1}^{n2} \left({}^{s}v_{j'} - {}^{s}z_{j'}\right)^2 \tag{7} $$

The connection weights between the second and fifth layer can be estimated by minimizing this equation, fixing connection weights from the first to the second and from the fifth to the sixth layer.

2.3 SOM-Based Convolutional Neural Networks

We use in this paper a new type of network architecture in which the SOM and CNN are combined with each other.

As is well known, the SOM has been used to detect features in input patterns and, in particular, to visualize the features in multi-dimensional spaces. Thus, the SOM has an ability to capture rich information in input patterns and to visualize over multi-dimensional spaces. However, this rich information has not necessarily been utilized for training supervised learning. This is mainly because the SOM can produce multi-dimensional feature spaces, and the conventional supervised neural networks cannot extract important information represented over multi-dimensional spaces (Fig. 3).

At this point, we propose here to combine the SOM with the CNN, because the CNN has been developed to deal with two- or three-dimensional data of images. The convolutional neural network is described, following the architecture used in the experimental results discussed below. In the architecture, we have three convolutional layers and one max-pooling layer. All these layers are supposed to be black-boxed.

As mentioned above, we use the bi-directional approach to estimating the black-box layers in the CNN component. First, the forward method is based on the SOM, Let ${}^s\mathbf{x}$ and $\mathbf{w}_j$ denote input and weight column vectors; then, distance between input patterns and connection weights is

$$\|{}^s\mathbf{x} - \mathbf{w}_j^{(2)}\|^2 = \sum_{k=1}^{L} ({}^s x_k - w_{kj}^{(2)})^2. \tag{8}$$

The ${}^s c$th winning neuron is computed by

$${}^s c = \mathrm{argmin}_j \, \|{}^s\mathbf{x} - \mathbf{w}_j^{(2)}\|. \tag{9}$$

The SOM tries to update connection weights into the winners and their neighbors.

The forward method is based on the outputs from the SOM, which can be computed by

$${}^s v_{j'}^{(2)} = \exp\left(-\frac{\|{}^s\mathbf{x} - \mathbf{w}_j^{(2)}\|^2}{\sigma}\right) \tag{10}$$

On the contrary, the backward method inversely computes the output of the original network. Then, the output from the seventh layer is computed by

$${}^s z_j^{(7)} = f\left(\sum_{i=1}^{n_8} w_{ij}^{(8)} {}^s t_i\right) \tag{11}$$

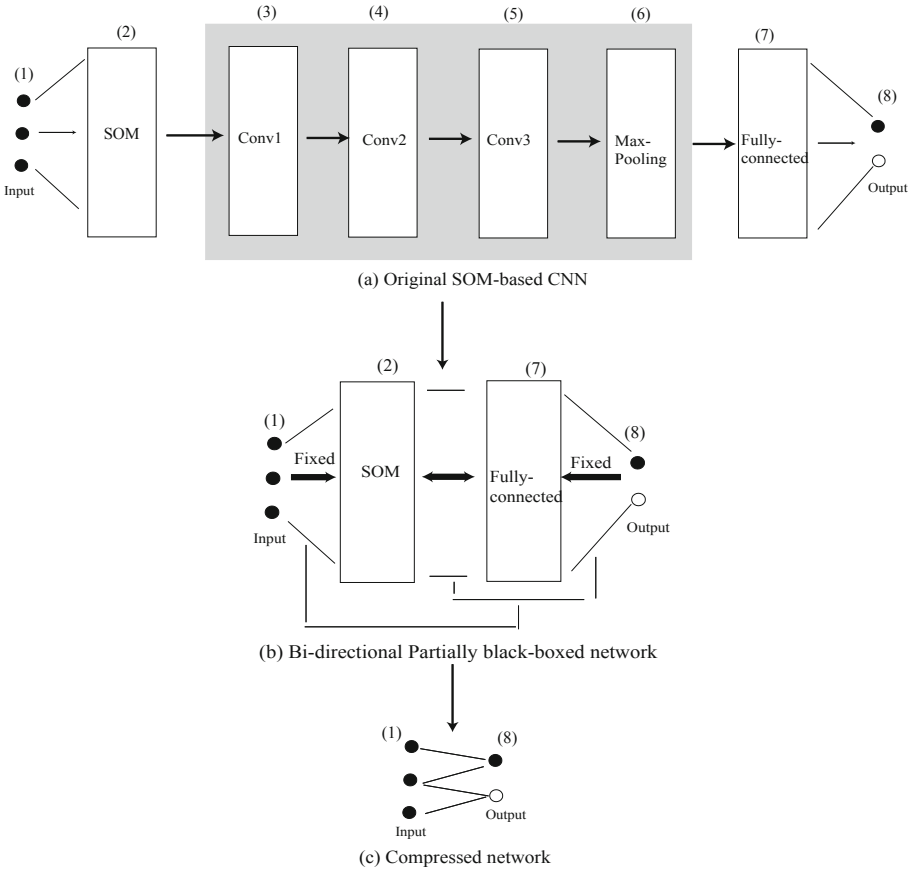


Fig. 3. SOM-based convolutional network by partially black-boxed operation to be compressed into the simplest one.

Then, the output from the second layer is computed by

$$ {}^{s}z_{j'}^{(2)} = f\left(\sum_{j=1}^{n_7} w_{j'j}^{(7)} {}^{s}z_{j}^{(7)}\right) \tag{12} $$

The target for the second layer should be the output from the SOM component. The final error is computed by

$$ E = \sum_{s=1}^{q}\sum_{j'=1}^{n2}\left({}^{s}v_{j'}^{(2)} - {}^{s}z_{j'}^{(2)}\right)^2 \tag{13} $$

The connection weights between the second and fifth layer can be estimated by minimizing this equation, fixing connection weights from the first to the second and from the fifth to the sixth layer.

3 Results and Discussion

3.1 Pulsar Data Set

Experimental Outline. The first experiment used the HTRU2 or pulsar data set [20], where we tried to classify input patterns into pulsar and non-pulsar ones. The original number of pulsar candidate patterns was 16,259, of which only 1,639 were actually pulsar ones. Thus, for easy comparison of the final results, we chose the even number of pulsar and non-pulsar ones, and the total number was reduced to 3,000. The attributes were the mean of the integrated profile, standard deviation of the integrated profile, excess kurtosis of the integrated profile, skewness of the integrated profile, mean of the DM-SNR curve, standard deviation of the DM-SNR curve, excess kurtosis of the DM-SNR curve, and skewness of the DM-SNR curve.

The number of inputs was nine, and the SOM mapped those inputs into a two-dimensional space with a 20-by-20 size. The SOM used the built-in SOM package in the Matlab. The outputs from the SOM were given into the CNN component in which there were three convolutional layers, and the filter size was 3 by 3 or 5 by 5, and the number of filters increased from 10 to 100. The outputs from the convolutional layers were transformed by the leaky rectified linear activation function. The final max pooling had the rectangular size of 2 by 2. All learning parameters were set to the default ones except the use of the leaky rectified linear activation and the validation data set with the 3 patience number for the present results to be easily reproduced.

SOM and Generalization. Figure 4 shows outputs from the SOM component. As can be seen in the figure, pulsar patterns were located on the upper side of the feature map in Fig. 4(a). On the other hand, non-pulsar patterns were located on the lower side of the feature map in Fig. 4(b), though non-pulsar patterns on the lower side were more widely spread over the feature space. Because the SOM can successfully classify the pulsar candidate patterns by representing the patterns over a two-dimensional feature space, this can contribute to the improvement of supervised learning.

Figure 5(a) shows generalization errors by the present method, bagging ensemble method, and logistic regression method. As mentioned above, those two conventional methods were used because they can produce the importance of input variables. As can be seen in the figure, the generalization errors by the present method were mainly lower than those by the two conventional methods. The lowest generalization error of 0.0563 was by the present method with 30 filters and a filter size of 3 by 3. The second lowest error of 0.057 was produced by the present method with 100 filters and a filter size of 5 by 5. The bagging ensemble method produced the error of 0.0592, the second worst generalization error. Finally, the logistic regression produced the worst error of 0.0599. These results confirmed that the SOM knowledge represented over a two-dimensional feature space and analyzed by the CNN component can be used to increase generalization performance.

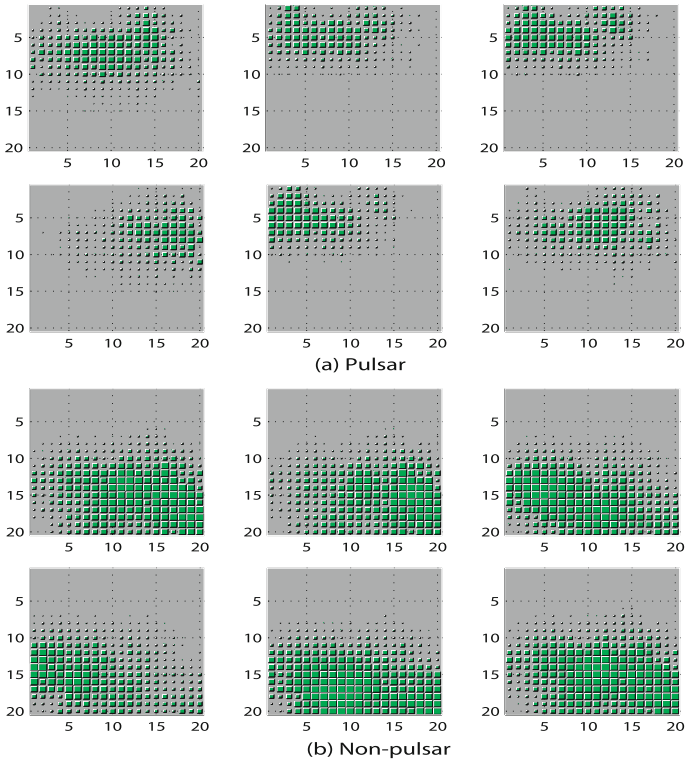


Fig. 4. Outputs from the SOM component, representing the existence of pulsar (a) and non-existence of pulsar (b).

Collective Interpretation. Then, we try to interpret the final connection weights by the present method, comparing the results with those by two conventional methods and the original correlation coefficients of inputs and targets.

Figure 5(b) shows the correlation coefficients between the correlation coefficients between inputs and targets of the original data set and weights strength of the present method, bagging method, and logistic regression and weights strength of compressed networks without knowledge or weights from the CNN component. The present method with 100 filters and the filter size of 3 by 3 produced the highest correlation coefficient of 0.9793. In addition, the present method with the filter size of 5 by 5 and 100 filters produced the second highest of 0.9758. This means that the compressed weights represent well the original correlation coefficients between inputs and targets. The compressed network without knowledge or weights from the CNN component produced the correlation coefficient of 0.1610. We can see that the weights or knowledge from the CNN could contribute to the performance of compressed networks, and the compressed weights have a possibility to represent the original weights of original SOM-based CNN networks. The bagging ensemble method produced the

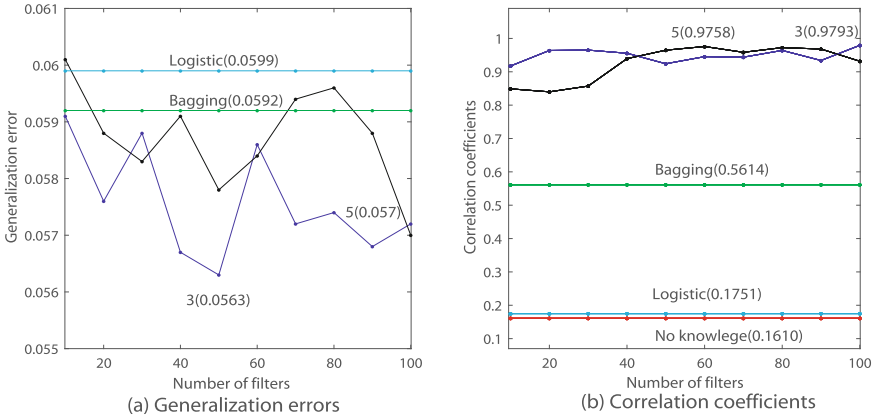


Fig. 5. Generalization errors by the present method, bagging, and logistic regression method (a) and correlation coefficients (b) between the original coefficients of inputs and targets and weights by the present method, importance by the bagging method and regression coefficients by the logistic regression analysis.

correlation coefficient of 0.5614, and the logistic regression analysis produced the second lowest coefficient of 0.1751. The logistic regression analysis could not produce stable results over different input patterns, and the final average coefficients were obtained in one sense by canceling out the individual coefficients for different input patterns.

Figure 6 (a) shows the correlation coefficient between inputs and targets of the original data set. When the knowledge or weights from the CNN were not injected in Fig. 6(b), the compressed weights were far from the correlation coefficients of the original data set in Fig. 6(a). When the bagging ensemble method was applied in Fig. 6(c), input No. 3 had the highest importance. The logistic regression analysis showed also that input No. 3 was the most important. Finally, the mutual information Fig. 6(e) showed a completely different result from the correlation coefficients. However, the most important input was input No. 1, which was also the largest absolute coefficient of the original correlation coefficients in Fig. 6(e). As explained in the reference paper [20], this mutual information may represent the characteristics of input patterns, but it is quite difficult to state that the results can represent input patterns. In addition, because mutual information cannot represent the negative effect of importance as is the case with the importance of the bagging ensemble method, the absolute importance of the correlation coefficients in Fig. 6(a) were not so different from the corresponding mutual information in Fig. 6(e). In the mutual information, importance gradually decreased from left to right, and in the original correlation coefficients in Fig. 6(a), the absolute strength of correlations gradually decreased from left to right. These results show that the original correlation coefficients between inputs and targets cannot be fully represented by the conventional methods as well as the simple estimation and method without knowledge from the CNN.

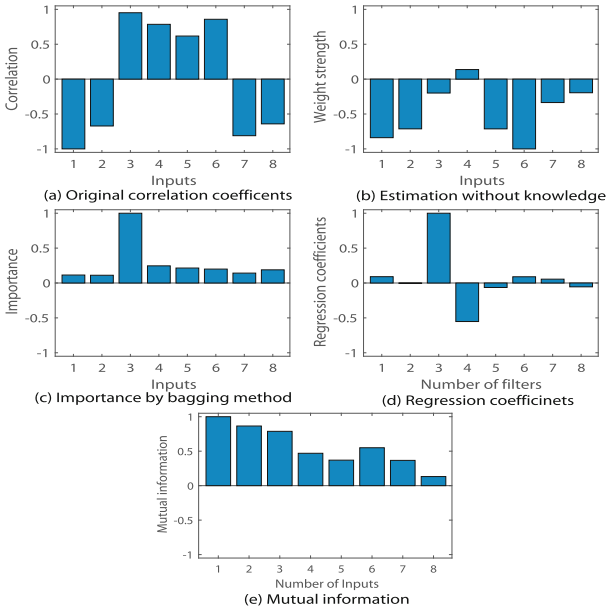


Fig. 6. The original correlation coefficients (a), weights without knowledge (b), importance by the bagging method (c), regression coefficients by the logistic regression analysis (d), and mutual information between inputs and outputs for the pulsar data set.

Figure 7 shows the compressed weights when the number of filters increased from 10 (a) to 100 (j). As can be seen in the figure, compressed weights were quite similar to the correlation coefficients of the original data set in Fig. 6(a). As explained in Fig. 5, the correlation coefficients between connection weights and the original correlation coefficients between inputs and targets were over 0.9 for almost all numbers of filters; it is natural that connection weights by all different filters showed weights similar to the original correlation coefficients.

Figure 8 shows the compressed weights averaged over 10 different initial conditions and 10 different filter numbers. As can be seen in the figure, the average weights were quite similar to the correlation coefficients, but the inputs in the middle decreased gradually, which is different from the correlation coefficients of the original data set.

3.2 Occupancy Data Set

Experimental Outline. The second experiment tried to classify inputs into an occupied or non-occupied state by five input variables: temperature, humidity, light, CO2, and humidity ratio [4]. The number of patterns in the training data set was 8,143, of which only 1,729 were considered occupied. For easy comparison, we used the even number of occupied and non-occupied patterns, and the actual

number of patterns was reduced to 3,000 with 1,500 occupied and non-occupied states. We used the Matlab deep learning package, and the defaults parameters were used as much as possible for the present results to be able to be reproduced easily.

SOM and Generalization. Figure 9 shows the outputs from the SOM component. As can be seen in the figure, the occupancy states were represented on the upper right-hand side of the feature map in Fig. 9(a). On the other hand, the non-occupancy states were represented on the lower left-hand side of the feature map. Those two types of input patterns seem to be classified by the diagonal boundary. Those feature maps can be expected to contribute to the improvement of generalization performance.

Figure 10(a) shows generalization by four methods. As can be seen in the figure, the generalization errors by the present method with filter sizes of 3 by 3 and 5 by 5 gradually decreased when the number of filters increased. When the number of filters was 100, the present method with a 3-by-3 filter size produced the lowest error of 0.0064. The filter size 5 by 5 produced even a smaller error of 0.0057 when the number of filters was 90. The bagging ensemble method produced good performance with the error of 0.0067, close to that by the present method with a 3-by-3 filter size. The logistic regression analysis produced the worst error of 0.0081.

Collective Interpretation. Collective weights were quite similar to the correlation coefficients between inputs and targets of the original data set. Figure 10 (b) shows correlation coefficients between the original correlation coefficients between inputs and targets and connection weights by the present method, importance by the bagging method, and the regression coefficients by the logistic regression analysis. The present method with 3 by 3 produced the correlation coefficient of 0.9759, the largest coefficient. The 5-by-5 filter size produced the second largest coefficient of 0.9576. The bagging ensemble method also produced the larger value of 0.8956. On the contrary, the compressed network obtained without the knowledge or connection weights by the CNN component produced the lower coefficient of −0.1238. Finally, the logistic regression analysis produced the coefficient of −0.2138.

Figure 11(a) shows the correlation coefficients between inputs and targets of the original data set. As can be seen in the figure, the third input of "light" showed the largest importance, and the second one was "CO2," followed by the third one of temperature, while the input variables of "humidity" and "humidity ratio" did not play an important role in classification. Figure 11(b) shows compressed weights without knowledge or weights from the CNN component. As can be seen in the figure, the first input showed much larger importance, while all the others took the smaller strength. As shown in Fig. 11(c), the bagging ensemble method produced an importance that was very close to the correlation coefficients in Fig. 11(a). Finally, the regression coefficients by the logistic regression analysis were different from the correlation coefficients in Fig. 11(d).

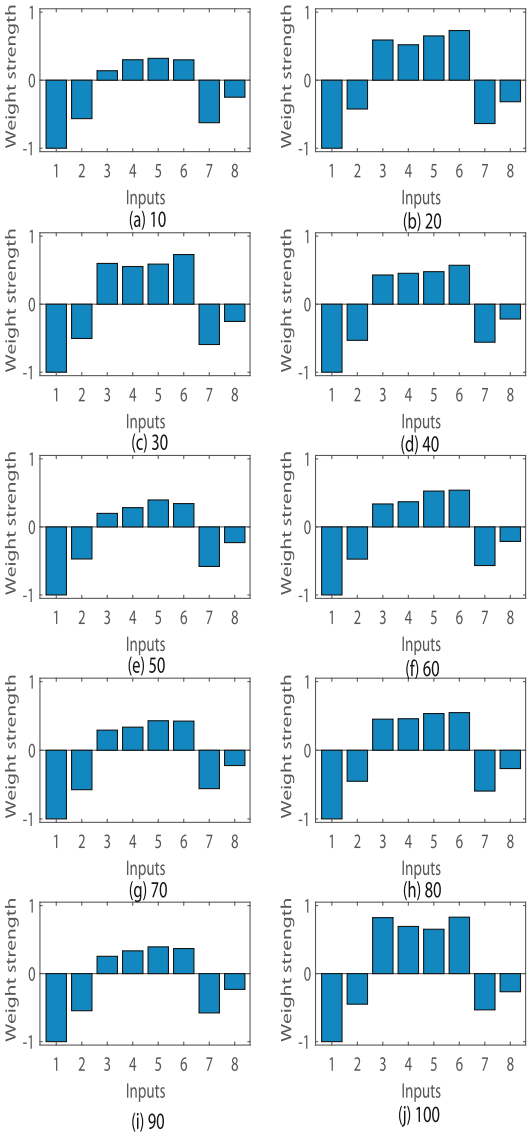


Fig. 7. Connection weights in the compressed networks when the number of filters increased from 10 (a) to 100 (j).

Figure 12 shows the compressed weights by the method with a 5-by-5 filter size when the number of filters increased from 10 to 100. When the number of filters was 10 in Fig. 12(a), differences in the strength among six input variables were relatively small. Then, gradually, the compressed weights became close to the original correlation coefficients, making input variable No. 3 stronger.

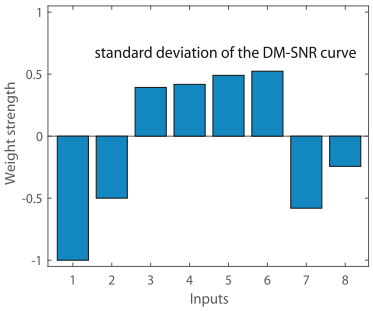


Fig. 8. Averaged weights over 10 initial conditions and 10 different numbers of filters.

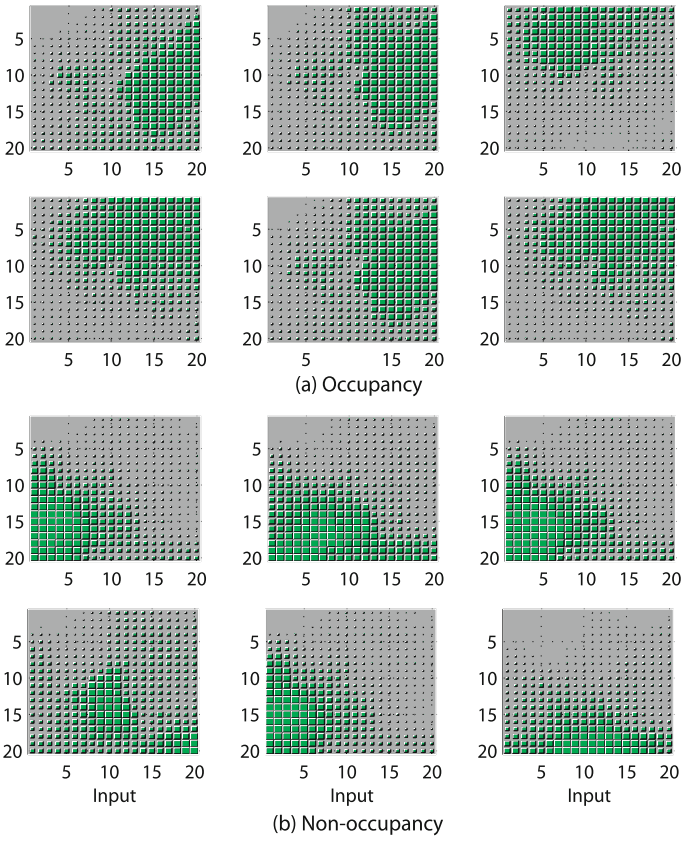


Fig. 9. The outputs from the SOM for the occupancy (a) and non-occupancy (b).

Figure 13 shows the average compressed weights over 10 different initial conditions and 10 different numbers of filters. As can be seen in the figure, the compressed weights were very close to the correlation coefficients in Fig. 11(a).

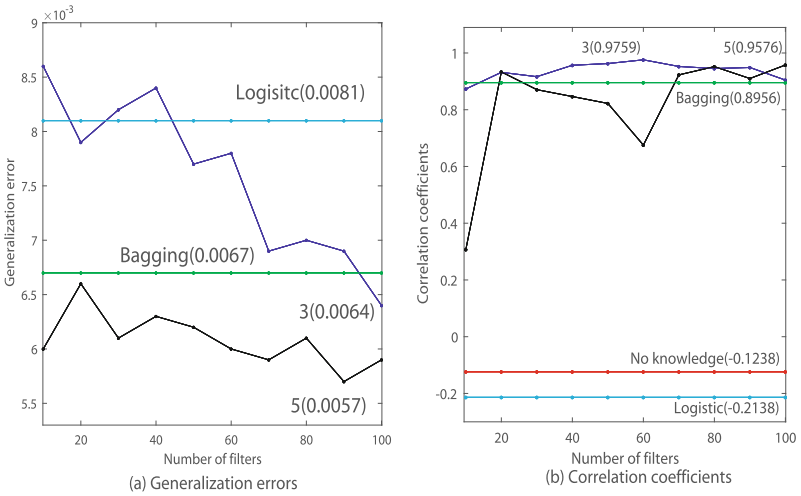


Fig. 10. Generalization errors by the present method, bagging, and logistic regression analysis (a) and correlation coefficients between the original coefficients and their corresponding weights and coefficients (b) for the occupancy data set.

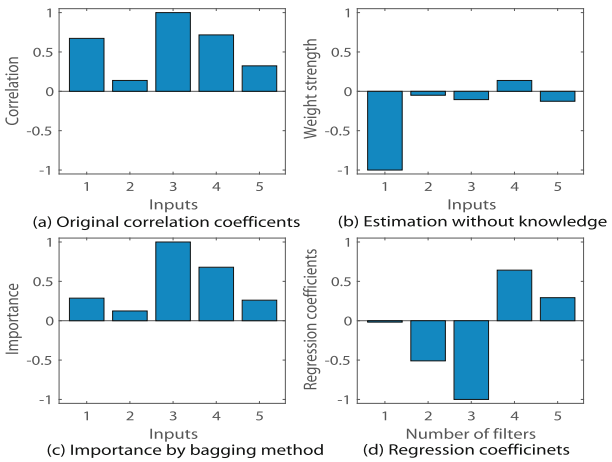


Fig. 11. Correlation coefficients (a), weights without knowledge (b), importance by the bagging method (c), and regression coefficients (d) by the logistic regression analysis for the occupancy data set.

As mentioned, the bagging ensemble methods could produce smaller errors, close to those by the present method. In addition, the importance of the bagging method was close to the correlation coefficient. This explains why the present method

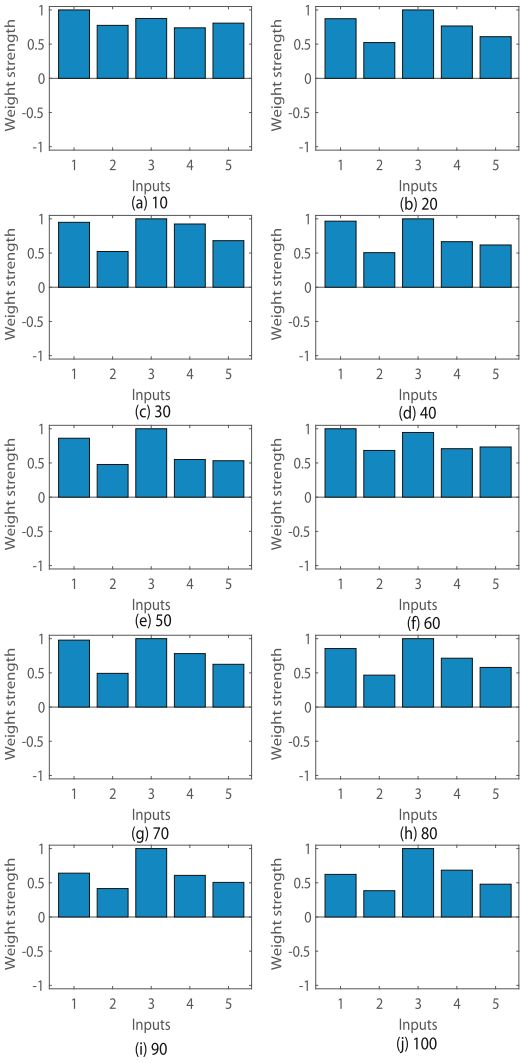


Fig. 12. Connection weights when the number of filters increased from 10 (a) to 100 (j) for the occupancy data set.

and the bagging method showed better generalization performance, because both methods could capture linear relations between inputs and targets clearly. The results obtained by the present experimental results were very close to those explained in the original paper dealing with the occupancy data set [4].

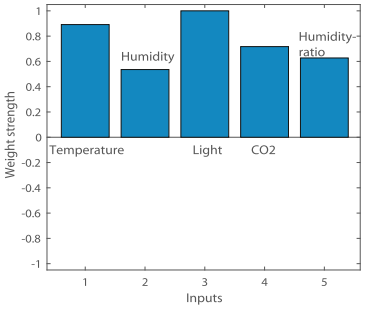


Fig. 13. Weights averaged over 10 different initial conditions and 10 different numbers of filters.

4 Conclusion

The present paper proposed a new type of interpretation method for complex and multi-layered neural networks. We have developed an interpretation method called "neural compressor" to compress multi-layered neural networks into the simplest ones without hidden layers. However, we have found that there are some cases where it is impossible to compress directly the complex multi-layered neural networks. In those cases, we have proposed a new method in which some parts with difficulty in compression are temporarily black-boxed, and then connection weights in those layers are estimated by examining the knowledge in the other parts. Thus, the method is called the "partially" black-boxed interpretation model. For this model, we proposed the bi-directional method, where the black-boxed parts are estimated by knowledge from the targets as well as from the inputs.

The method was applied to the SOM-based convolutional neural networks. The SOM has been extensively used in many applications, in particular for visualization of complex data over two-dimensional feature spaces. The rich knowledge created by the SOM has produced many supervised learning methods or hybrid methods to incorporate the SOM knowledge to improve generalization performance. However, those methods have not necessarily produced successful results. This is because the conventional methods cannot deal with the two-dimensional feature spaces created by SOM. Thus, we proposed a method here to combine the SOM with the CNN in one framework. However, it is very hard to compress the CNN component into the simplest one, and we black-box the CNN component. Then, we make a multi-layered neural network with the black-boxed components for the CNN component. To estimate the inference mechanism of the black-box components, we proposed the bi-directional method to estimate the black-boxed parts of original multi-layered neural networks.

The method was applied to two well-known data sets, namely, pulsar and occupancy detection. In both data sets, improved generalization performance was obtained by the present method. In addition, the final compressed weights were very close to correlation coefficients between inputs and targets of the original data sets.

Several problems should be pointed out. For the estimation of black-boxed components, we developed the bi-directional approach, which tries to acquire information from targets as well as inputs as much as possible. However, information from targets is processed not through the inverse activation function but through the ordinary activation function. Thus, we need to weaken the fixed connection weights from the CNN component to adjust connection weights for the new activation function. More study is needed to simplify the bi-directional method for easy implementation. The other problem is that the number of neurons in the CNN component increases rapidly when the number of layers increases. Then, the number of neurons in the estimation networks correspondingly grows with heavy computation even for small data sets. Thus, we need to reduce the number of neurons in the CNN component before the black-box operation is applied.

Finally, one of the main future directions is the interpretation of the black-boxed component of CNN. We need to interpret fully the inference mechanism of the method and to interpret the black-boxed components.

References

1. Adebayo, J., Gilmer, J., Goodfellow, I., Kim, B.: Local explanation methods for deep neural networks lack sensitivity to parameter values. arXiv preprint arXiv:1810.03307 (2018)
2. Afolabi, M.O., Olude, O.: Predicting stock prices using a hybrid Kohonen self organizing map (SOM). In: 40th Annual Hawaii International Conference on System Sciences, HICSS 2007, pp. 48–48. IEEE (2007)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
4. Candanedo, L.M., Feldheim, V.: Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. Energy Buildings **112**, 28–39 (2016)
5. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. In: Advances in Neural Information Processing Systems, pp. 6967–6976 (2017)
6. Datar, M., Qi, X.: Automatic image orientation detection using the supervised self-organizing map. In: Proceedings of the 8th IASTED International Conference on Signal and Image Processing. Imaging Technologies (2006)
7. De Runz, C., Desjardin, E., Herbin, M.: Unsupervised visual data mining using self-organizing maps and a data-driven color mapping. In: 16th International Conference on Information Visualisation (IV), pp. 241–245. IEEE (2012)
8. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal, vol. 1341, no. 3, p. 1 (2009)
9. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a right to explanation. arXiv preprint arXiv:1606.08813 (2016)
10. Ismail, S., Shabri, A., Samsudin, R.: A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time-series forecasting. Expert Syst. Appl. **38**(8), 10574–10578 (2011)
11. Jin, F., Qin, L., Jiang, L., Zhu, B., Tao, Y.: Novel separation method of black walnut meat from shell using invariant features and a supervised self-organizing map. J. Food Eng. **88**(1), 75–85 (2008)

12. Jirapummin, C., Wattanapongsakorn, N., Kanthamanon, P.: Hybrid neural networks for intrusion detection system. In: Proceedings of ITC-CSCC, vol. 7, pp. 928–931 (2002)
13. Kamimura, R.: Internal and collective interpretation for improving human interpretability of multi-layered neural networks. Int. J. Adv. Intell. Inf. **5**(3), 179–192 (2019)
14. Kamimura, R.: Neural self-compressor: collective interpretation by compressing multi-layered neural networks into non-layered networks. Neurocomputing **323**, 12–36 (2019)
15. Kaski, S., Nikkila, J., Kohonen, T.: Methods for interpreting a self-organized map in data analysis. In: Proceedings of European Symposium on Artificial Neural Networks. Bruges, Belgium (1998)
16. Kindermans, P.J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (Un) reliability of saliency methods. In: Explainable AI Interpreting Explaining and Visualizing Deep Learning, pp. 267–280. Springer (2019)
17. Kohonen, T.: Self-Organization and Associative Memory. Springer, New York (1988)
18. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)
19. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Networks **8**(1), 98–113 (1997)
20. Lyon, R.J., Stappers, B., Cooper, S., Brooke, J., Knowles, J.: Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. Mon. Not. R. Astron. Soc. **459**(1), 1104–1123 (2016)
21. Mao, I., Jain, A.K.: Artificial neural networks for feature extraction and multivariate data projection. IEEE Trans. Neural Networks **6**(2), 296–317 (1995)
22. Ohno, S., Kidera, S., Kirimoto, T.: Efficient automatic target recognition method for aircraft SAR image using supervised SOM clustering. In: 2013 Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), pp. 601–604. IEEE (2013)
23. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: why did you say that? arXiv preprint arXiv:1611.07450 (2016)
24. Shieh, S.L., Liao, I.E.: A new approach for data clustering and visualization using self-organizing maps. Expert Syst. Appl. **39**(15), 11924–11933 (2012)
25. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
26. Su, M.C., Chang, H.T.: A new model of self-organizing neural networks and its application in data projection. IEEE Trans. Neural Networks **123**(1), 153–158 (2001)
27. Titapiccolo, J.I., Ferrario, M., Cerutti, S., Barbieri, C., Mari, F., Gatti, E., Signorini, M.: A supervised SOM approach to stratify cardiovascular risk in dialysis patients. In: XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013, pp. 1233–1236. Springer (2014)
28. Tsai, C.F., Lu, Y.H.: Customer churn prediction by hybrid neural networks. Expert Syst. Appl. **36**(10), 12547–12553 (2009)
29. Varshney, K.R., Alemzadeh, H.: On the safety of machine learning: cyber-physical systems, decision sciences, and data products. Big data **5**(3), 246–255 (2017)
30. Vesanto, J.: SOM-based data visualization methods. Intell. Data Anal. **3**, 111–126 (1999)
31. Wu, S., Chow, T.W.: PRSOM: a new visualization method by hybridizing multidimensional scaling and self-organizing map. IEEE Trans. Neural Networks **16**(6), 1362–1380 (2005)

32. Xu, L., Chow, T.: Multivariate data classification using PolSOM. In: Prognostics and System Health Management Conference (PHM-Shenzhen), 2011, pp. 1–4. IEEE (2011)
33. Xu, L., Xu, Y., Chow, T.W.: PolSOM-a new method for multidimentional data visualization. Pattern Recognit. **43**, 1668–1675 (2010)
34. Xu, Y., Xu, L., Chow, T.W.: PPoSOM: a new variant of polsom by using probabilistic assignment for multidimensional data visualization. Neurocomputing **74**(11), 2018–2027 (2011)
35. Yin, H.: ViSOM-a novel method for multivariate data projection and structure visualization. IEEE Trans. Neural Networks **13**(1), 237–243 (2002)

PrivLeAD: Privacy Leakage Detection on the Web

Michalis Pachilakis[1,2], Spiros Antonatos[1], Killian Levacher[1], and Stefano Braghin[1(✉)]

[1] IBM Research, Dublin, Ireland
[2] University of Crete/FORTH, Heraklion, Greece
stefanob@ie.ibm.com

Abstract. Each person generates a plethora of information just by browsing the Web, some of which is publicly available and some other that should remain private. In recent years, the line between public and private/sensitive information is becoming harder to distinguish and the information generated on the Web is being sold and used for advertising purposes, turning the personal lives of users into products and assets. It is extremely challenging for users and authorities to verify the behavior of trackers, advertisers and advertising websites in order to take action in case of misconduct. To address these issues, we present PrivLeAD, a domain independent system for the detection of sensitive data leakage in online advertisement. Specifically, PrivLeAD leverages Residual Networks to analyze the advertisements a user receives during normal internet browsing and thus detect in real time potential leakage of sensitive information. The system has been tested on real and synthetic data proving its feasibility and practical effectiveness.

Keywords: Online privacy · Web advertisement · Transparency

1 Introduction

Billions of users browse the web to read news, connect with their friends via social networks, buy products from online stores and more. This online activity generates a huge amount of data which, in the current information era, has proven to be the most valuable resource. [18]. In fact, several companies have built their whole business model on user data aggregation and utilization, which they later provide, for a fee, as a service. The larger and most commonly known of such ecosystem is advertising. When users start browsing the web, trackers that are present in each visited website, will try to construct the online persona of each user via various techniques [11,13]. Publishers use the collected information to target users with more personalized advertisements, so that they can increase their revenue in order to cover the cost of their utilities and make some profit. On the other hand, users continue to use their favorite online services for free and also receive advertisements tailored to their interests and needs, which could

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 428–439, 2021.
https://doi.org/10.1007/978-3-030-55180-3_32

be proven useful. This two way relationship is the gas that fuels the web and is proven beneficial for both parties. A problem arises when trackers begin to gather information about sensitive topics, such as medical conditions, political views, and sexual orientation. Most users would consider these pieces of information private and would not be willing to share or publicly disclose them. Furthermore, leakage of such sensitive information could publicly expose, or in some cases even harm, users. For example, insurance companies might decline services, based on information collected from third parties, which might indicate an all alleged previous medical condition.[1]

With the establishment of legislation like General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) users are entitled to more control over their data, but it is still unclear how well publishers comply with such legislation. Additionally, it is extremely challenging for users and authorities to verify the behavior of advertising websites or advertisers in order to take action in case of misconduct.

In this paper our motivation is to enhance transparency on the sensitive data trackers and advertisers collect. We aim to uncover how the private and personal information of a user are used by third parties. We therefore propose a first of its kind methodology that reveals information leakage by inferring the content delivered to the user. We design and implement Privacy Leakage Detection (PrivLeAD), a system for the detection of privacy leakage in online advertisements that allows users to define sensitive information policies and notifies them when misuse of their data is observed. This follows the work of other researches, such as [3] where the authors describe the necessity of tools to create awareness among Internet users about the monetary value connected to the commercial exploitation of their online personal information.

The rest of the paper is organized as follows. Section 2 presents the mechanics behind tracking and online advertising and defines our motivation. Section 3 presents in detail our systems architecture while Sect. 4 describes the building blocks from the methodological point of view. Section 5 explores the feasibility of our system and provides our experimental results. Section 6 discusses the related work and finally in Sect. 7 we summarize our contribution.

2 Motivation

Tracking is a technique used to collect information about users based on specific patterns or habits. During the last decade data companies have been in a continuous spree to obtain larger and more accurate data. Currently, most websites use several elements of JavaScript code which are executed each time a user loads a page. This code is responsible for reporting several pieces of information, such as the visited web-page, user identifiers, geolocation, browser information, and more, to a third party company. These third party companies, namely *trackers*, try to collect information about the users, reconstruct their browsing history in

[1] https://www.forbes.com/sites/jessicabaron/2019/02/04/life-insurers-can-use-social-media-posts-to-determine-premiums.

order to reveal user interests and habits, and then sell this information for profit to anyone interested. The effectiveness of a tracker is defined by its reach, as in how many web-pages, and by its collaborations with other trackers. Currently 95% of web-pages contain at least one tracker, 78% of which try to send unsafe data [20].

Trackers are central to the Real-Time Bidding (RTB) ecosystem. RTB is a programmatic way to sell advertisement ad-slots, i.e. placeholders inside a website where advertisement are rendered, to the highest bidder of a real time auction. Every time a user visits a website utilizing RTB, advertisers compete for an ad-slot. The advertiser that offers the highest bid wins the auction, which subsequently renders its advertisement. RTB accounts for more than 74% of the online advertising [14]. RTB is a complex and closed ecosystem with several entities competing and collaborating. We briefly present the most important of them:

- **Publisher**: the owner of a website and where advertisement auctions take place. Each publisher reserves placeholders (ad slots) where the ads will be rendered.
- **Advertisers**: the buyers of the ad slots. They create ad campaigns and define the audience they want to target.
- **Supply-Side Platforms (SSPs)**: an entity that allows the publisher to manage its ad-slots, define minimum prices willing to sell them, and so on.
- **Ad-exchange (ADX)**: a real time market place, similar to a stock exchange that allows advertisers and publishers to sell and buy ad slots.
- **Demand-Side Platforms (DSPs)**: an agency that utilizes sophisticated algorithms to help advertisers draw better bidding decisions and buy better ad-slots.

In order for RTB to work, advertisers need to obtain user information to tune their bidding policies and to decide if they are interested in specific users. Trackers take upon the task of collecting information about the users in order to sell it to advertisers. Without trackers, advertisers would not be able to target users making advertisement campaigns less effective and thus generating less revenue. Motivated by the relationship between trackers and the RTB entities, we try to infer sensitive data leakage on the web. To do that, we leverage the mechanics behind online advertising and the intuition that, in order for users to get specific advertisements, the advertisers need to collect some knowledge about them. Hence, we can infer how sensitive information are exploited in the wild by observing the type of served advertisements.

3 System Architecture

PrivLeAD consists of two main components and an ancillary one. Namely, a web browser plugin (the Client), a collection of remote services (the Server) and, optionally but strongly recommended, the Obfuscation Proxy.

3.1 PrivLeAD Browser Plugin

The plugin runs on the user's browser and it is responsible for retrieving all the advertisements that the user receives when browsing. At the same point, the plugin sends the identified advertisements to the server. The plugin's design has been driven by the following requirements:

- **Privacy by Design**: It must not expose user's identity or compromise her/his security.
- **Real time operations**: It must promptly report any leakage to the user.
- **Lightweight**: It should not change the user experience by consuming a significant amount of resources such as processor, memory or bandwidth.
- **Transparency**: All the operations performed by the plugin should be transparent to the user.

The plugin leverages the *webRequest* API to collect all of the images received by the browser. As we focus only on advertisements, we need to exclude images related to the website in order to reduce the traffic with the server. For this reason, we exclude all of the images originating from the website's domain. Additionally, we created a list with Content Delivery Networks (CDNs) serving images. Before sending the images for inspection to the server, we check if they are served by a known CDN, in which case we exclude them as not advertisements (see Sect. 4.1). We also manage a cache of the alleged advertisements that a user sees per session, to further reduce the traffic.

On the "front-end" side, PrivLeAD provides to the user a friendly and easy to use UI. It allows the definition of policies, characterizing what topics, or combination of, are considered sensitive and in what context. Furthermore, it provides reports about the topics associated with each identified advertisement and it alerts the user for policy violations. Note that the policies are stored and validated only locally, reducing the chances of information leakage cause by the plugin. PrivLeAD is written if a few hundred lines of JavaScript code, and it currently supports Chrome and Firefox browsers.

3.2 PrivLeAD Server

The server acts as the heart of our system. It is responsible for further distinguishing advertisements from images and detecting the topics correlated to each advertisement. Although the client component tries to remove all non advertisements, we observed that this is not enough. For this reason we further classify the images upon arrival on the server, as advertisement or non-advertisement by utilizing the residual network for advertisement classification described in Sect. 4. Following this task, we proceed to extract the topics from the advertisements. Again, we use the models described in Sect. 4 to achieve this outcome. When the classification is finished, we aggregate the results and send them back to the client. On the client side, the plugin checks if the identified topics match any of the ones that the user defined as sensitive. In such cases, it notifies her about possible sensitive information leakage highlighting the violating advertisement.

3.3 PrivLeAD Obfuscation Proxy

When performing the information leakage detection, we must ensure that the system does not leak any information itself. Even without sending Personal Identifiable Information (PII), an honest but curious server would be able to recreate user sessions by observing specific patterns such as frequency of requests or information like the IP address of a user. Using this information, the server would be able to identify specific users and their online habits based on the advertisements they receive. To address this problem, we introduce a middle proxy. The proxy is responsible for removing the real IP address before sending the image to the server for the analysis, as well as shuffling the received images from several different users. This way we leave the server in an agnostic state because it only sees the proxy's IP address. Hence, it cannot infer user sessions based on frequency patterns. Once deployed at scale, we envision the PrivLeAD proxy assigned to globally trusted organization to further enhance the trust in the system.

Note that similar functionality can be obtained by leveraging systems like TOR[2], either by configuring the browser to use or by just redirecting the traffic generated by the plugin. The main purpose of suggesting the utilization of a proxy is to simplify the users' experience while utilizing PrivLeAD, while aiming at high privacy standards.

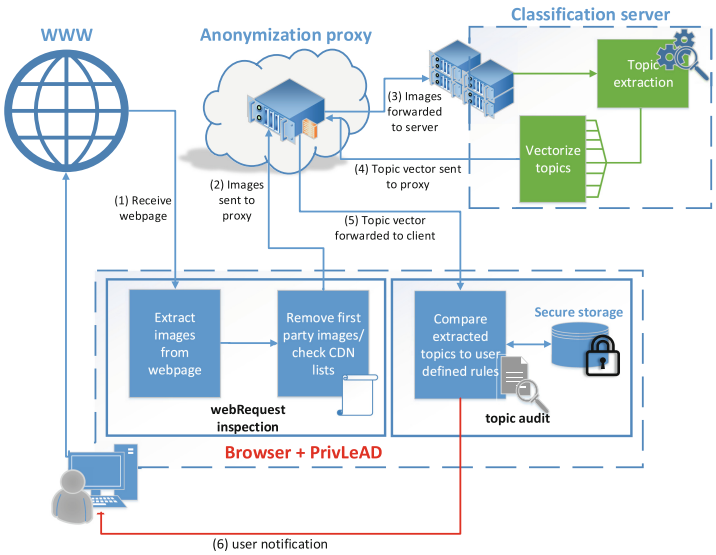


Fig. 1. High level architecture of PrivLeAD

[2] https://www.torproject.org/.

3.4 PrivLeAD Operational Workflow

Figure 1 shows a high level overview of all the components and their interactions. When a user visits a web-page and starts receiving the content (1), the PrivLeAD plugin will send all the alleged advertisements to the proxy (2). In turn, the proxy will forward the request to the server (3) hiding the user's identity. Once an advertisement is forwarded to the server, more advanced classifications are applied to reduce false positives, and if the advertisement is confirmed a such, a score for each topic is computed. These "raw" scores are combined and reasoned upon to identify higher level topics that are returned to the proxy (4) and then back to the client (5). The client validates the extracted high level topics against the user's preferences also leveraging further contextual information and it alerts the user in case of potential violations (6).

4 Methodology

As we already mentioned, our goal is to detect and report back to the user any kind of leakage that could be considered as a privacy breach. We do not aim to minimize the privacy leakage as other systems already do [15,16]. Rather, we provide a supplementary role, by informing the users of what kind of data may have been leaked and which advertisements leverage them to be more relevant and appealing. Our approach is advertisement-centric and leverages the following idea: *In order for an advertisement to be presented to a user, the advertiser must have some kind of knowledge related to the user.* Which is the building block of the real time bidding advertising ecosystem [12–14,20].

In order to be able to detect private data leakage we need to identify 1) the advertisements in each web page and 2) the topic (or topics) of each advertisement. In what follows, we describe in more detail how the required classification is performed.

4.1 Advertisement Classification

Firstly, we need to classify which of the images are advertisements and which are not. We obtained the dataset of [6] which consists both of advertisement and non-advertisement images. Similarly to their approach, we trained a residual network [5] with 50 layers using $8,348$ advertisements and $13,597$ non-advertisements. After tuning the hyper-parameters of the model and the optimizer, we achieved 89% precision on a corpus of $2,000$ images of the previous dataset. We also achieved 86% precision on a dataset containing $18,000$ images collected by automatic website crawling.

PrivLeAD is meant to be domain independent and, thus, it embodies an agnostic approach. This means that we try to detect all the advertisements and not only for well known advertisers. This generates false positives, where images may be classified as advertisements. To tackle this problem we investigated two possible solutions. The first one involves the creation of a blacklist containing

known CDNs that prevents content from those sources to be sent to the server. The second approach consists of the definition of empirical rules for not processing images with easily measurable characteristics that are not similar to known advertisements. As a first approach we use image sizes, both in terms of pixels and bytes.

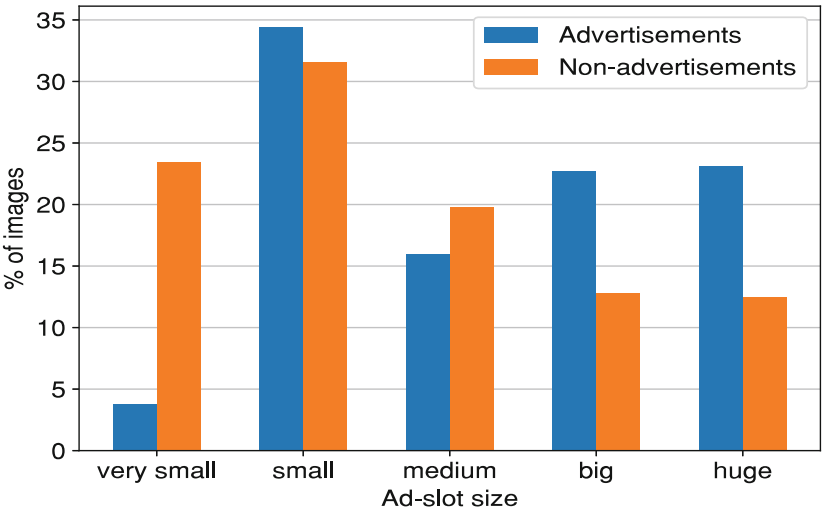


Fig. 2. Sizes of advertisement images.

We analyzed the images classified as advertisements by the model in terms of their URL and size. Figure 2 shows the sizes of the images grouped in 5 categories defined as follows: *very small* ($\leq 80 \times 80$ px), *small* ($\leq 300 \times 250$ px), *medium* ($\leq 500 \times 400$ px), *big* ($\leq 800 \times 600$ px) and *huge*. By simply removing the *very small* images false positives drop 24%, with only a 4% increment in false negatives. Furthermore, after analyzing the image URLs we were able to refine the CDN, reducing false positives by 26% with no impact on false negative.

4.2 Topic Classification

After classifying an image as advertisement we need to extract the topic (or topics) relevant to it. Again, we leveraged the dataset provided by [5]. The authors have already classified the images into 39 categories. From those categories, we could distinguish 16 sensitive categories, 6 of which had enough samples to properly train a model. We followed a one-vs-all approach to create models for each category for two reasons. First, it is faster and easier to train. When we need to add a new category we just need to train a model for it and not re-train the whole model for each category. Second, by having multiple smaller models the system is more scalable as the classification can be executed in parallel by faster

models. Therefore, we decided to train a distinct 32 layers residual network for each of the 6 categories, namely *alcohol*, *financial*, *health*, *law regulations*, *travel*, and *shopping habits*.

5 Experimental Evaluation

To test and validate PrivLeAD we used both real and carefully crafted personas of specific interests.

We created the personas following the approach described in [17]. Namely, we designed 6 personas, each focused on a different area among *education*, *financial*, *gaming/gambling*, *healthcare*, *shopping*, and *travel*. For each persona we identified a set of web pages that were used for training. The training pages are of similar interests for each persona and they had active ad campaigns. We developed a crawler, based on *Selenium*, that simulates the browsing behavior by scrolling on the page, clicking on internal links and introducing random latency between the various actions. Note that the crawler accepts privacy popups and stores cookies to better simulate a real user. The training activity aims at correlating each synthetic persona to a specific interest. After the training, we visit a set of control pages. The control pages consist of *neutral* web pages, such as weather websites. Between the visits of the control pages, we introduce at random visits to some of the training pages to reduce the noise potentially generated from control pages. When crawling with the personas, we collected all the images available and the correlated metadata, such as the image's URL, the publisher's domain and a timestamp of the collection. Note that these information are not collected in the real implementation of PrivLeAD. Nonetheless, during testing phase this helped in refining the list of CDN used to reduce the false positives, as described in Section 4.1, and to better understand the characteristics of the studied ecosystem.

We collected between few hundred to tens of thousand images per day over a month, to a total of 243,452 images. We separated the images based on the day they where collected and the persona they correspond. After removing the duplicate images, we ran an analysis to correctly distinguish between actual advertisements and the rest of the images. Figure 3 shows the percentage of advertisements received by each persona. Note that each persona received a different percentage of advertisements, with *shopping* and *health* being the most targeted.

Beside collecting and identifying advertisements, the server needs to be able to detect which topics are expressed in the advertisements themselves. To correctly understand the accuracy of our topic extraction models, we used the advertisements received from the personas and we classified them with on our topics extraction module. After that, we leveraged human annotators to validate the correctness of the classification. In order for the classification to be considered correct, the majority of the human annotators need to agree with the result from our model [6]. Figure 4 shows the results from the model annotation. Each advertisement can, and should, be classified with more than one topics but

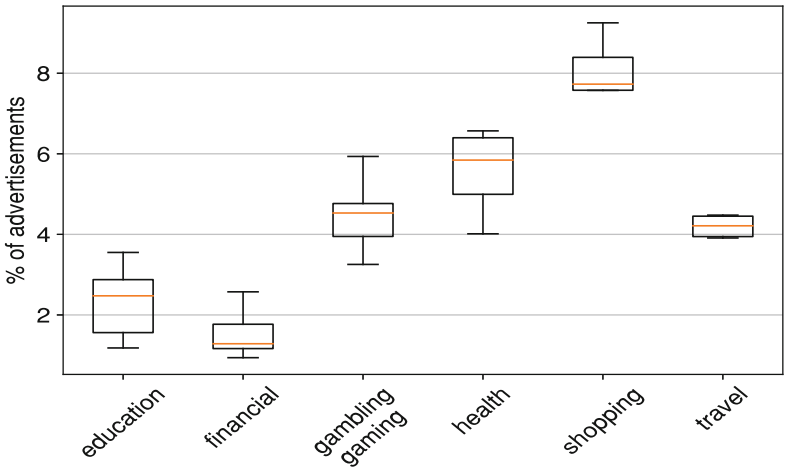


Fig. 3. Percentage of advertisements per persona.

for clarity reasons we plot only the dominant topic. The topic *travel* has the highest amount of advertisements across the personas, followed by *finance* and *healthcare*. The experiment took place during summer, we are therefore not surprised that advertisements about traveling and vacation are the most popular. The smallest percentage is, surprisingly, observed for advertisement primarily classified as *shopping*. One would expect *shopping* to hold a much higher percentage. This is the result of plotting only the dominant topic. After further analysis of the results of the topic classification, we indeed observed that *shopping* is a strong secondary topic. This is the reason why, when reporting the

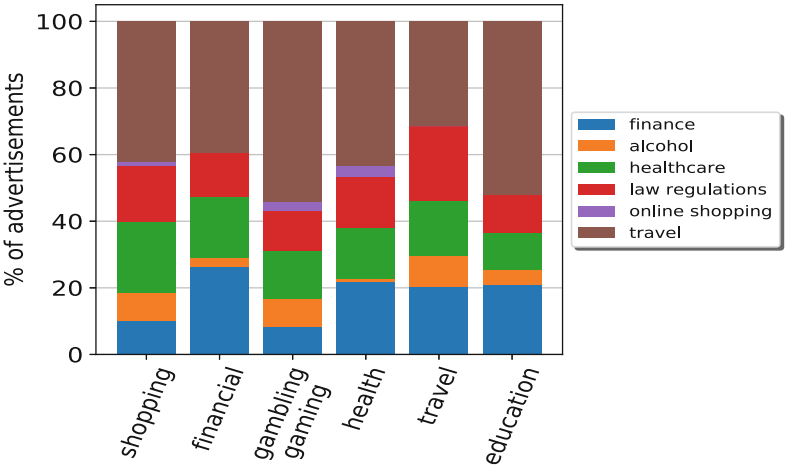


Fig. 4. Sensitive topics classified per persona.

topic to the users PrivLeAD displays all the sensitive topics found ranked by confidence. This would help the users to better understand about the semantic of the advertisements. For example, an advertisement classified as *alcohol* and *shopping* implies "buying alcohol", while *alcohol* and *law regulations* imply "awareness when drinking and driving".

We extracted a large sample of the classified images and we gave it to human annotators. The human annotators agreed in 61.6% of the cases with the predictions of the models.

5.1 Discussion

The empirical evaluation of the system clearly demonstrates the feasibility of the approach and the utility of the tool itself. Our analysis identified areas that need improvement and that constitute corollary work to what is here presented. First of all, the identification of advertisements is based on a set of heuristics and trained models. Both components can be improved to further increase accuracy by simply expanding the training set used. We are continuously collecting for new examples of advertisements with the final goal to produce a larger silver standard corpus. Also, topic classification can be improved. The results presented are sufficient to prove feasibility and satisfactory from a practical point of view. Nonetheless, there is room for improvement either by providing better training material or exploring new model options for specific topics. On this front, it is also useful to the set of available topic categories. This is a straightforward operation given the server's architecture but it requires more advertisements to achieve good accuracy.

6 Related Work

Data leakage on the web is a well studied problem over the years. In [10] the authors propose a two speed system, allowing users to browse the web without sharing sensitive information within social networks, and to share private information only when required. The proposed system leverages a sandbox around the Facebook login button and it can be adapted to other social networks. On the other hand, [19] proposes an architecture that enables targeting without compromising user privacy. In this approach behavioral profiling and targeting takes place in the user's browser instead of the online service's side. In [8] the authors present an empirical study of privacy-violating information flows, that is an important class of web vulnerability. The authors implement a flow engine within the Chrome browser to detect privacy violations such as cookie stealing, location hijacking, history sniffing, and more. In [9] the authors presents a framework to detect and to categorize the privacy violations in online social networks. Dual to our approach, [4] presents a system that privately delivers advertisements to users. The system delivers to the client a set of advertisements given a single interest and generic demographic information. The client system selects which advertisements to show based on private information stored locally on

the user's machine. Thus, an advertiser does not access the user's private information. In [1] the authors investigate the advertisements on Facebook and they suggest that it commercially exploited potentially sensitive personal data for advertising purposes through the ad preferences that it assigns to its users. In a recent study [7] in 136 mental health websites it was discovered that 98.78% of them contained third party elements and 76.04% contained third-party trackers for marketing purposes. It was even observed that in some cases user's data was directly shared with trackers and advertisers. Tools like [2,15,16] have been developed to enhance online privacy by blocking trackers and advertisers. The here proposed approach is not a competitor of these tools, contrary PrivLeAD complements them by providing more transparency to the final users.

7 Conclusion

In this paper, we presented PrivLeAD, a novel tool for assisting users in the identification of the misuse of sensitive data and proper privacy violation. We designed and implemented our system for both Chrome and Firefox browsers, to prove its feasibility. We described the technical details relative to our tool and give guidelines that can be used by systems sharing a similar design. The effectiveness of PrivLeAD's approach has been demonstrated with both real and synthetic users, where we were able to identify advertisements with high precision and to correctly identify their semantics.

References

1. González Cabañas, J., Cuevas, Á., Cuevas, R.: Unveiling and quantifying facebook exploitation of sensitive personal data for advertising purposes. In: 27th USENIX Security Symposium (USENIX Security 18) (2018)
2. Electronic Frontier Foundation. Privacy badger. https://www.eff.org/privacybadger
3. González Cabañas, J., Cuevas, Á., Cuevas, R.: FDVT: data valuation tool for facebook users. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI 2017. ACM (2017)
4. Guha, S., Cheng, B., Francis, P.: Privad: practical privacy in online advertising. In: USENIX 2011 (2011)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR, abs/1512.03385 (2015)
6. Hussain, Z., Zhang, M., Zhang, X., Ye, K., Thomas, C., Agha, Z., Ong, N., Kovashka, A.: Automatic understanding of image and video advertisements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
7. Privacy International. Your mental health for sale. Technical report (2019)
8. Jang, D., Jhala, R., Lerner, S., Shacham, H.: An empirical study of privacy-violating information flows in Javascript web applications. In: Proceedings of the 17th ACM conference on Computer and Communications Security (2010)

9. Kökciyan, N., Yolum, P.: PriGuard: a semantic approach to detect privacy violations in online social networks. IEEE Trans. Knowl. Data Eng. **28**, 2724–2737 (2016)
10. Kontaxis, G., Polychronakis, M., Markatos, E.P.: Minimizing information disclosure to third parties in social login platforms. Int. J. Inf. Secur. **11**, 321–332 (2012)
11. Mayer, J.R., Mitchell, J.C.: Third-party web tracking: policy and technology. In: IEEE Symposium on Security and Privacy (2012)
12. Pachilakis, M., Papadopoulos, P., Markatos, E.P., Kourtellis, N.: A measurement study of the header bidding ad-ecosystem, No more chasing waterfalls (2019)
13. Papadopoulos, P., Kourtellis, N., Markatos, E.P.: Cookie synchronization: everything you always wanted to know but were afraid to ask. CoRR, abs/1805.10505 (2018)
14. Papadopoulos, P., Rodríguez, P.R., Kourtellis, N., Laoutaris, N.: If you are not paying for it, you are the product: how much do advertisers pay to reach you? In: Proceedings of the 2017 Internet Measurement Conference, IMC (2017)
15. Parmar, A., Dedegikas, C., Toms, M., Dickert, C.: Adblock plus efficacy study. Accessed 24 Oct 2015
16. Signanini, J.M., McDermott, B.: Ghostery (2014). https://www.ghostery.com/
17. Solomos, K., Ilia, P., Ioannidis, S., Kourtellis, N.: Cross-device tracking: systematic method to detect and measure CDT. CoRR, abs/1812.11393 (2018)
18. The Economist. The world's most valuable resource is no longer oil, but data (2017)
19. Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H., Barocas, S.: Adnostic: privacy preserving targeted advertising. In: Proceedings Network and Distributed System Symposium (2010)
20. Yu, Z., Macbeth, S., Modi, K., Pujol, J.M.: Tracking the trackers. In: The 25th International Conference on World Wide Web (2016)

A Neuro-Fuzzy Model for Software Defects Prediction and Analysis

Riyadh A. K. Mehdi(✉)

Ajman University, Ajman, UAE
r.mehdi@ajman.ac.ae

Abstract. Identifying defective software modules is a major factor in creating software in a cost effective and timely manner. A variety of machine learning techniques are available for predicting defective software modules. This paper investigates the effectiveness of using a fuzzy neural network in identifying faulty modules and the relative importance of the various software metrics in predicting defective modules and their role in explaining the presence of software defects. The conclusions of the work is that the model provides good accuracy but low probability of detecting software defects. However, the model does provide useful insight into the predictive and explanatory power of the various metrics affecting defect detection. The work shows that the most influential predictive metrics are those related to the measurement of program length and complexity. However, the degree of their relative influence is different for different datasets. On the explanatory side, metrics relating to program complexity, program length, and number of operators and operands are the most influential in explaining the presence of software defects. Again, their relative importance and the importance of other metrics vary for different datasets. These results are tentative and influenced to some degree by the model architectural parameters and configurations. Future work will concentrate on tuning the model and identifying a group of software characteristic that collectively can provide a good probability of defect detection.

Keywords: Neuro-fuzzy inference · Software defects prediction · Neural networks · Fuzzy systems

1 Introduction

Identifying defective software modules is critical to the success of software projects in terms of functionality and can have huge impact on the time and cost of developing and maintaining the software. Software modules require different amount of testing efforts and resources, consequently defect prediction models can play a vital role in focusing testing resources on the most likely defective software components. Research efforts in software defect prediction attempt to build models using software metrics to train the models and use the models to predict defects on newly developed software. These efforts has contributed to the research on software metrics as well as the development of machine learning algorithms for this purpose [1].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 440–451, 2021.
https://doi.org/10.1007/978-3-030-55180-3_33

The problem of detecting software defects have been approached from two different perspectives. The first is to design a suitable set of relevant software features and metrics, and the second is to develop prediction models based on data mining and machine learning techniques [2]. Halstead [3] developed a set of software metrics such as program length, program volume, difficulty level, and line count software metrics based on operators and operands count. McCabe [4] developed a set of software metrics based on program complexity such as cyclomatic complexity, essential complexity, design complexity among others. Chidamber and Kemerer [5] developed a CK metrics suit for object-oriented design measures based on inheritance depth, coupling between classes, and methods' cohesion. As for defect prediction, a variety of algorithms have been proposed using random forests, naive Bayes, support vector machines, decision trees, and neural networks [6].

A cost-effective software defect prediction model should have low false negatives (faulty modules not detected) so that modules with high probability of containing errors are identified and can be inspected first [7]. This issue becomes even more important for safety critical systems where software errors cannot be tolerated in any component of the system.

2 Literature Review

Cui et al. [8] investigated the effect of the number of predictors on the performance of eight different defect prediction models based on AUC values obtained for each model. They used the correlation coefficient as an indicator of model stability. They concluded that the performance of the C4.5 model is highly dependent on the number of features used and that the Sequence Minimum Optimization (SMO) model is the least affected. The performance of other models was highly dependent on the dataset used with change in the number of features used having little effect.

Rodrigues et al. [9] have studied the problem of imbalanced data in datasets and its effect on software defect prediction models' performance. They used two different versions of NASA datasets obtained from the PROMISE repository and the other a cleaned and preprocessed version prepared by Shepperd et al. [10]. Their results indicated that predictive models that deal with imbalanced data achieve higher identification rates of defects. They concluded that preprocessing of the datasets for errors and duplicates, characteristics of the datasets, degree of data imbalance, and the prediction model itself played a significant role in the classifications results [10].

Okutan and Yildiz [11] developed a Bayesian network model to determine the probabilistic dependencies among software metrics. They used the Promise repository metrics in addition to two metrics they defined: number of developers, and lack of code quality. The model has been designed to learn the marginal defect probability of the whole software system, the most significant metrics, and the causal relationships among metrics [11]. Their experiments on nine Promise datasets indicated that response for a class, lines of code, and lack of coding quality constitute the most likely indicators of defective software. While, coupling, weighted methods per class and lack of cohesion are the least effective metrics in identifying faulty modules. They also showed that the number of children, and depth of the inheritance tree have a marginal effect in addition to being

unreliable. In addition, they showed that the probability of detecting a software fault becomes higher as the number of programmers increases [11].

Maheshwari and Agrawal [12] developed a two-stage defect prediction model for object oriented software based on the premise that defect prediction is a cost-sensitive classification problem where misclassification cost consists of the cost of testing none-defective modules and the cost of missing on defective modules. The model uses Random Forest classification ensemble and three-way decision process. In the first stage, modules are divided into three categories: defective, non-defective, and differed. Differed modules are further classified in the second stage. They reported that their three-way decision model gives better results in terms of classification error and decision cost compared with a two-way decision model.

Gupta et al. [13] constructed a stepwise regression model to assess the relevance of CK (Chidamber and Kemerer), Halstead, and object-oriented metrics to the identification of different defect categories to reduce testing costs. Pearson correlation was used to identify the relationship between metrics suites and defect categories. They reported that reuse ratio, lack of cohesion in methods, packages imported, and response for a class metrics are good predictors of correctness defect. Lack of cohesion in methods, reuse ratio, and average cyclomatic complexity are best for bad practice defect. Unweighted class size, LCOM2, instance variable declared, number of lines of code, weighted methods per class, Halstead effort, response for a class, and LCOM are the best predictors for performance defect with message passing coupling, and reuse ratio as best metrics for unreliable code defect. They used structural equation modeling to validate the results of their prediction model.

Lu et al. [14] investigated the performance of a semi-supervised random forest algorithm for software fault prediction. They used embedded multidimensional scaling strategy to reduce the dimensional complexity of software metrics. They concluded that the same semi-supervised learning algorithm with dimension-reduction as a preprocessing step performs significantly better than without preprocessing in situations where few modules with known defective content are available for training. They pointed out that their approach is particularly suitable in cases where the number of software modules available for model training is low.

Petric et al. [15] built an ensemble of classifiers from different families using a stacking approach for predicting faulty modules. Their results showed an improved performance compared to the most commonly used bagging technique. They also stated that a stacking ensemble does not require many base classifiers as long as the classifiers are diverse and come from different families. They also stated that Naïve Bayes and Sequential Minimum Optimization seems to work well in combination.

Mehdi [16] has investigated the use of radial basis function and probabilistic neural networks in software defect prediction. The work has shown that these types of neural networks provide an acceptable level of accuracy but a poor probability of detecting a faulty module. However, probabilistic neural networks performed consistently better than radial basis functions using the Promise datasets. The conclusion of the investigation is to use a range of software defect prediction models to complement each other rather than relying on a single technique.

This paper investigates the effectiveness of using an adaptive neuro fuzzy inference system based on the Sugeno fuzzy model to predict software defects. In addition, the work also examines the relative importance of the metrics affecting software defect prediction.

3 Research Methodology

3.1 Adaptive Neural Network Based Fuzzy Inference System

Neural networks are computational mechanisms that can solve problems requiring human intelligence by learning from data, while fuzzy systems are concerned with making inferences in the presence of vagueness and ambiguity based on a set of fuzzy rules provided by human experts [17]. A shortcoming of fuzzy systems is the lack of ability to learn and adapt to changes in their environment. The opposite is true with neural networks, they can learn but their reasoning is embedded in the connection weights of their neurons [18]. The integration of fuzzy inference systems and neural networks can provide a platform to build intelligent systems [17] by replacing the weakness of one system by the strength of the other [17]. To realize this integration, Jang [19] proposed a neural network that is functionally equivalent to a Sugeno fuzzy inference model called the Adaptive Neural Network Based Fuzzy Inference System (ANFIS) that integrates the mechanisms of neural networks and fuzzy inference systems to utilize the capabilities of both.

3.2 ANFIS Architecture

Jang's ANFIS [19] is normally represented by a six-layer feedforward neural network. Figure 1 shows the ANFIS structure that corresponds to a first-order Sugeno fuzzy model with two inputs and two membership functions per input [17]. The network has two types of neurons: fixed, represented as a circle and adaptive depicted as a square. The following exposition is adapted from [17].

For a first-order Sugeno fuzzy model, a two-rule rule base is expressed as follows:

1. If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$
2. If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

Let the membership function of each fuzzy set Ai, and Bi for i = 1,2 be defined by a Gaussian membership function μ_F as follows:

$$\mu_{A_i}(x) = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^{2b_i}} \tag{1}$$

In evaluating the rules, a product T-norm (logical and) is chosen. Evaluating the rule premises using product T-norm results in,

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y), \ \mathrm{i} = 1, 2. \tag{2}$$

Evaluating the implication and the rule consequents gives,

$$f(x, y) = \frac{w_1(x, y)f_1(x, y) + w_2(x, y)f_2(x, y)}{w_1(x, y) + w_2(x, y)}. \tag{3}$$

Leaving the arguments out,

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \tag{4}$$

The above equation can be rewritten as,

$$f = \overline{w_1} f_1 + \overline{w_2} f_2, \text{ where}$$
$$\overline{w_i} = \frac{w_i}{w_1 + w_2} \tag{5}$$

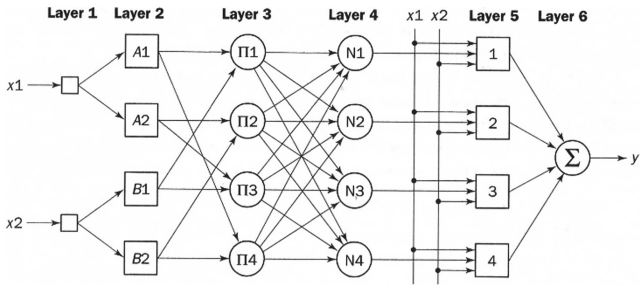


Fig. 1. An adaptive Sugeno neuro-fuzzy inference system Architecture.

3.3 ANFIS Learning

ANFIS uses a combination of least-squares estimator and the gradient descent algorithm to learn its parameters [19]. Initially, an activation function is assigned to each neuron representing a membership function. The centers of the membership functions are set so that the range of an input is divided equally and the widths and slopes are set to allow sufficient overlapping of the respective membership functions. For each epoch, training is conducted in two steps: forward pass and a backward pass. In the forward pass, the ANFIS learning mechanism uses training patterns to estimate the parameters of the rules' consequents by a least-squares algorithm. Once the rules' consequent parameters are established, the network can compute the error. In the backward pass, the errors are propagated backward and the parameters of the membership functions are adjusted using the back-propagation learning algorithm [17]. In the ANFIS training algorithm suggested by Jang [19], both antecedent parameters and consequent parameters are optimized. In the forward pass, the consequent parameters are adjusted while the antecedent parameters remain fixed. In the backward pass, the antecedent parameters are modified while the consequent parameters are kept fixed. When the input-output data set is relatively small, membership functions can be described by a human expert and kept fixed throughout the training process [17].

3.4 Programming Environment

To implement the neuro-fuzzy system, the following Matlab functions were used [20]:

1. *genfis3*(*Xin, Xout, type, clusterNum, fcmOptions*), *genfis3* generates a fuzzy inference system using fuzzy c-means clustering by extracting a set of rules that models the data behavior with *fcm ()*. The function *fcm ()* determine the number of rules and membership functions for the antecedents and consequents. The arguments for *genfis3* are as follows [20]:

   - *Xin*: a matrix where each row contains the input values of a data point. The matrix *Xin* have one column per input.
   - *Xout*: a matrix where each row contains the output values of a data point. The matrix *Xout* has one column per output.
   - *type*: has two options, '*mamdani*' or '*sugeno*'.
   - *clusterNum*: number of clusters to be generated by *fcm()*. The number of clusters determines the number of rules that model the system's behavior. The value can be an integer or '*auto*'. When *clusterNum* is '*auto*', the function uses Matlab *subclust* algorithm with a radius of 0.5 and the minimum and maximum values of *Xin* and *Xout* to find the number of clusters.
   - *fcmOptions,* [*option (1) option (2) option (3) option (4)*]:
   - *option (1):* a parameter that controls the degree of fuzzy overlap between clusters. This value must be greater than 1, with smaller values creating more crisp cluster boundaries, default value is 2.0.
   - *option (2)*: Specify the maximum number of optimization iterations.
   - *option (3)*: Specify the minimum improvement in the objective function between successive iterations for the learning process to continue.
   - *option (4)*: a zero-one value that allows the user to choose whether to display the value of the objective function or not after each iteration.
     When omitting *fcmoptions*, the function uses the default values.

2. *anfis*(*trainingData, options*), This function fine-tune Sugeno-type fuzzy inference system using training data and options. It generates a single-output Sugeno fuzzy inference system (FIS) and tunes the system parameters using the specified input/output training data and options. The FIS object is automatically generated using grid partitioning. The *options* allow the user to specify an initial FIS object to tune; validation data for preventing overfitting to training data; training algorithm options; and whether to display training progress information. The training algorithm uses a combination of the least squares and backpropagation gradient descent methods to model the membership functions of the training dataset.
3. *output* = *evalfis*(*fis,input*) uses the fuzzy inference system *fis* with input values in *input* and returns the resulting output values in *output*.

3.5 Software Defect Prediction Datasets

The following NASA software defect prediction datasets available publicly from the PROMISE repository are used in this research [21]:

- KC1: a C++ system implementing storage management for receiving and processing ground data.
- KC2: same as KC1 but with different personnel.
- CM1: is a NASA spacecraft instrument written in "C".
- JM1: A real-time predictive ground system written in "C".
- PC1: is s flight software for earth orbiting satellite written in "C".

3.6 Performance Measures

In this work, accuracy and probability of detection are used for the evaluation of model performance. These statistics are computed from the confusion matrix. A confusion matrix is used to measure the performance of a classifier. Figure 2 illustrates a confusion matrix for a two-class classifier. A good classifier would have large true positives and true negatives, and small number of the other two statistics [22].

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Fig. 2. Confusion matrix parameters.

Based on the confusion matrix, accuracy and probability are computed as:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{6}$$

$$Probability\ of\ detection = TN/(FN + TN) \tag{7}$$

3.7 Results and Discussion

Table 1 shows the accuracy and predictive power of the neuro fuzzy model for the PROMISE datasets. Derived attributes were removed and only the following attributes were used as predictors:

1. McCabe's line count of code – LOC
2. Cyclomatic complexity – CYC
3. Essential Complexity – ECP
4. Design Complexity – DCP
5. Halstead total operators and operands – TOO
6. Halstead Effort – EFF
7. Halstead Line Count – HLC
8. Flow graph branch count – FBC

Table 1. Accuracy and prediction of the neuro fuzzy model.

Data set	Accuracy	Prediction
KC1	0.853	0.357
KC2	0.828	0.739
CM1	0.848	0.214
JM1	0.825	0.404
PC1	0.860	0.133

The results in Table 1 are compared with those shown in Table 2 obtained from a previous study using radial basis function and probabilistic neural networks [16]. On average, the performance of the model with regard to accuracy for the five datasets is slightly better than the accuracy shown in Table 2 for other neural networks based models. Probability of prediction is higher for all the datasets compared with those shown in Table 2.

Table 2. Accuracy and prediction of the RBNN and PNN.

	RBFNN		PNN	
Dataset	*Acc %*	*PD %*	*Acc %*	*PD %*
KC1	77.0	44	83.1	29.7
KC2	78.0	50	83.4	50.0
CM1	82.5	20	87.3	33.3
JM1	77.0	31	77.5	30.0
PC1	89.2	32	91.6	40.0

3.8 Relative Importance of Predictors

One of the main objective of this work is to investigate the effectiveness of the individual predictors used in the model in terms of their predictive and explanatory power of software defects.

3.9 Predictive Performance

To examine the predictive power of each explanatory input variable (metric), the neuro-fuzzy model was retrained by removing one input variable at a time and examining the predictive power of the model using test data for each dataset. Table 3 gives the predictive power of the model for each removed predictive input variable. The lower the resulting predictive power the model, the more effect the removed input variable has on the model predictability.

Table 3. Model predictive power resulting from removing one predictor at a time.

Data set	LOC	CYC	ECP	DCP	TOO	EFF	HLC	BCN
KC1 0.38	0.26	0.31	0.33	0.29	0.26	0.3	0.24	0.36
KC2 0.74	0.65	0.78	0.70	0.70	0.74	0.61	0.69	0.74
CM1 0.51	0.50	0.36	0.38	0.43	0.22	0.36	0.29	0.29
JM1 0.43	0.38	0.28	0.28	0.4	0.26	0.29	0.38	0.29
PC1 0.34	0.20	0.27	0.16	0.2	0.33	0.27	0.20	0.13

Table 3 shows that the number of lines of code (LOC, and HLC), total number of operators and operands (TOO), and program complexity as measured by BCN are the most influential input variables. This indicates that program length and program complexity are the most important predictors of software defects. However, the degree of their influence is different for different datasets, i.e. the type of software projects dataset represents.

3.10 Explanatory Performance

To identify the relative explanatory power of each input metric, sensitivity analysis was conducted to determine the causal importance of each predictor. For each dataset, each input variable was perturbed by steps of 5% from 0% to 100% at a time and the accuracy of the model was calculated. Changes in model accuracy for each input variable and dataset are shown in Fig. 3 to Fig. 7.

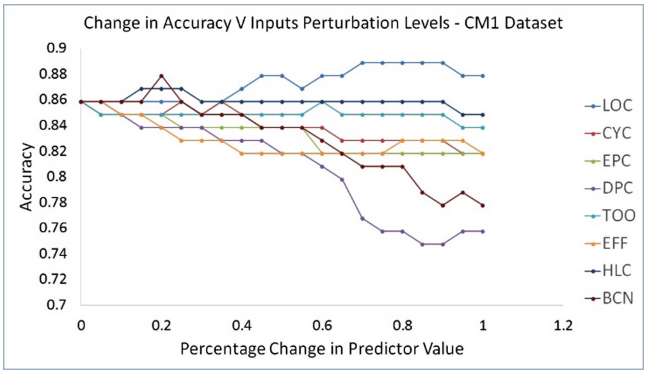


Fig. 3. Changes in accuracy as a function percentage increase in predictors for the CM1 dataset.

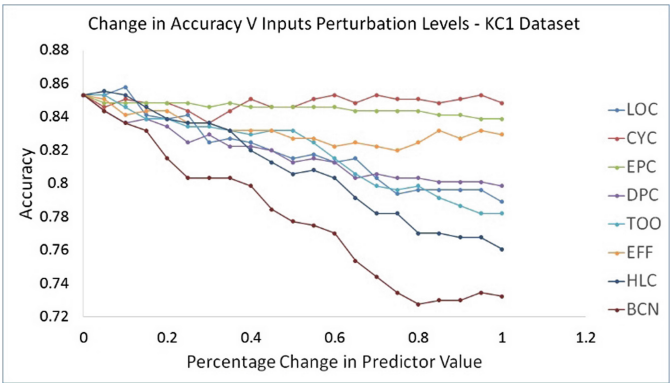


Fig. 4. Changes in accuracy as a function percentage increase in predictors for the KC1 dataset.

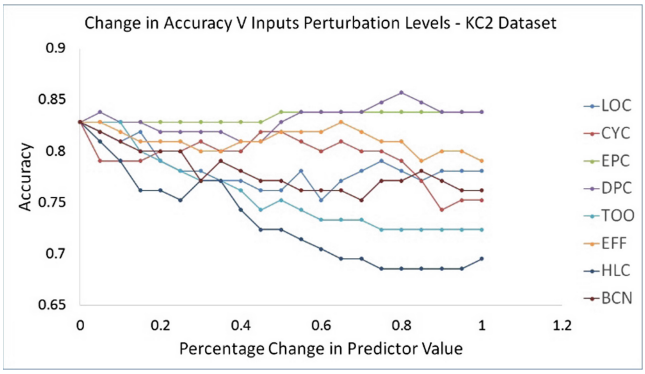


Fig. 5. Changes in accuracy as a function percentage increase in predictors for the KC2 dataset.

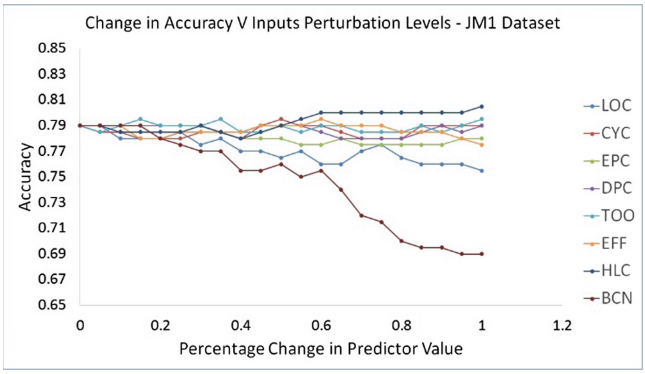


Fig. 6. Changes in accuracy as a function percentage increase in predictors for the JM1 dataset.

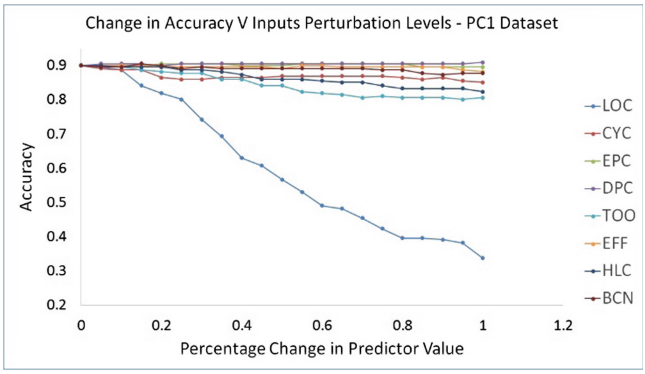


Fig. 7. Changes in accuracy as a function percentage increase in predictors for the PC1 dataset.

For the CM1 dataset, see Fig. 3, high degree of program complexity as measured by design complexity (DCP) and flow graph branch count (BCN) are the most influential in explaining the presence of software defects. Figure 4 shows that BCN is the most important input variable explaining the presence of software defects in KC1 dataset followed by Halstead number of lines of code (HLC). For the KC2 dataset, Fig. 5 indicates the program length (HLC) and number of operators and operands (TOO) are most important factors in explaining software defects. BCN is the most important single factor affecting the presence of software for the JM1 dataset, Fig. 6. For the PC1 dataset, software defects seem to be influenced by the number of lines of code (LOC) while the other input variables have no or little effect.

4 Conclusions

In this work, we have built a neuro fuzzy inference systems for detecting software defects. The model provides good accuracy but low probability of detecting software defects. However, the model does provide valuable insight into the predictive and explanatory power of the input metrics used in software defect detection. The work shows that the most important predictive input variables are those related to program length and program complexity. However, the degree of their influence is different for different datasets. On the explanatory side, factors relating to program complexity and program length and number of operators and operands are the most influential in explaining the presence of software defects. However, their relative importance and the importance of other factors varies for different datasets. These results are tentative and future work will concentrate on identifying the most influential group of factors that influence software defects detection.

References

1. Humphreys, J., Dam, H.K.: An explainable deep model for defect prediction. In: IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (2019)

2. Wang, S., Liu, T., and Tan, L.: Automatically learning semantic features for defect prediction. In: Proceedings of the 38th IEEE International Conference on Software Engineering, pp. 297–308 (2016)
3. Halstead, M.H.: Elements of Software Science (Operating and Programming Systems Series). Elsevier Science Inc., New York (1977)
4. McCabe, T.J.: A complexity measure. IEEE Trans. Softw. Eng. **SE-2**(4), 308–320 (1976)
5. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. IEEE Trans. Softw. Eng. **20**(6), 476–493 (1994)
6. Wu, X.-Y., Liu, L.: Dictionary learning-based software defect prediction. In: Proceedings of the 36th International Conference on Software Engineering, pp. 414–423 (2014)
7. Zhang, H., Cheung, S.C.: A cost-effective criterion for applying software defect prediction models. In: Proceedings of the 9th Joint Meeting on Foundations of Software Engineering, pp. 643–646 (2013)
8. Cui, M., Sun, Y., Lu, Y., Jiang, Y.: Study on the influence of the number of features on the performance of software defect prediction model. In: Proceedings of 3rd International Conference on Deep Learning Technologies, pp. 32–37 (2019)
9. Rodriguez, D., Herraiz, I., Hassison, R., Dolado, J., Riquelme, J.C.: Preliminary comparison of technologies for dealing with imbalance in software defect prediction. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (2014)
10. Shepperd, M., Song, Q., Sun, Z., Mair, C.: Data quality: some comments on the NASA software defect datasets. IEEE Trans. Software Eng. **30**(9), 1208–1215 (2013)
11. Okutan, A., Yıldız, O.T.: Software defect prediction using Bayesian networks. Empirical Softw. Eng. **19**(1), 154–181 (2012)
12. Maheshwari, S., Agrawal, S.: Three-way decision based defect prediction for object oriented software. In: Proceedings of the International Conference on Advances in Information Communication Technology & Computing (2016)
13. Gupta, N., Panwar, D., Sharma, A.: Modeling structural model for defect categories based on software metrics for categorical defect prediction. In: Proceedings of the Sixth International Conference on Computer and Communication Technology, pp. 46–50 (2015)
14. Lu, H., Cukic, B., Culp, M.: Software defect prediction using semi-supervised learning with dimension reduction. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, pp. 314–317 (2012)
15. Petric, J., Bowes, D., Hall, T., Christianson, B., and Baddoo, N.: Building an ensemble for software defect prediction based on diversity selection. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (2016)
16. Mehdi, R.: Software defect prediction using radial basis and probabilistic neural networks. Int. J. Comput. Appl. Res. **5**(5), 260–265 (2016)
17. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems. Addison Wesley, Harlow (2017)
18. Abraham, A., Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: a survey in soft computing framework. IEEE Trans. Neural Networks **2**(3), 748–768 (2000)
19. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993)
20. Matlab. https://www.mathworks.com/help/fuzzy/genfis3.html. Accessed 12 Oct 2019
21. Shirabad, S.J., Menzies, T.J.: The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada (2005). http://promise.site.uottawa.ca/SERepository
22. EMC Education Services, Data Science & Big Data Analytics. Wiley (2015)

Fast Neural Accumulator (NAC) Based Badminton Video Action Classification

Aditya Raj, Pooja Consul(✉), and Sakar K. Pal

Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India
pconsul@seas.upenn.edu

Abstract. Automatic understanding of sports is essential to improve viewer experience and for coaches and players to analyze, strategize and improve game performance. To achieve this it is essential to harness the ability to localize and recognize the actions in sports videos. In this paper we focus on the fast paced sport of badminton. The challenge is to extract relevant spatio-temporal features from several consecutive frames and to classify them as an action or a no-action in minimal time and with minimal computational power. We propose two novel Neural Accumulator (NAC) based frameworks, namely NAC-LSTM and NAC-Dense for aforementioned objective. Neural Accumulator is employed for spatial and temporal feature extraction respectively followed by classification. The actions of the players were annotated as *react*,*lob*, *forehand*, *smash*,*backhand* and *serve*. An Autoencoder-LSTM Network, Dilated Temporal Convolutional Networks (TCN) and Long Term Recurrent Convolutional Network (LRCN) have been designed for comparison. Multiclass recognition has been performed with 5-fold cross validation on several test-train data splits (from 10–50%) to verify the efficacy of the results. The proposed methods achieve a high classification accuracy in strikingly minimal CPU time.

Keywords: Neural Accumulator · Video action classification · Machine vision

1 Introduction

In the last decade, as the available sports multimedia has grown, the technology for analysis of content-based sports video has followed. Thereafter, due to high commercial potential and wide viewership, it has become paramount to develop a potent representation of sports video content (Zhu et al. [1]). Several investigations using traditional approaches relying on learning a frame based spatio-temporal features (Zhao and Elgammal [2]) for action recognition and motion tracking focusing on close-up view of human body parts motion have been carried out (Pingali et al. [3], Shah and Jain [4]). Such an approach is constrained with the requirement of high resolution and multiple near-view cameras.

A. Raj and P. Consul—Authors have equal contribution.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 452–467, 2021.
https://doi.org/10.1007/978-3-030-55180-3_34

In case of far-view frames, tracking human body motion for the purpose of action classification becomes less efficient and hence inapplicable. The advent of deep learning techniques in motion tracking enabled extracting robust spatio-temporal features with lesser constraint on the input. The success of Deep convolutional networks (CNNs) in visual and Recurrent neural networks (RNNs) in sequential data interpretation tasks, encouraged its utilization in video action recognition.

In this context, Donahue et al. [5] employed a temporal feature based deep learning scheme incorporating a recurrent neural network and conventional 2D-CNN with end to end training for activity recognition and captioning of videos. This is contrary to the learning scheme based on a frame by frame spatio-temporal representation by either pooling over motion features or time-varying weight learning. Single stream networks also utilized the prowess of 2-D CNNs for feature extraction from frames and fusing the information at different levels of the network (Karpathy et al. [6]). Tran et al. [7], assessed the merits of 3D-CNNs for spatio-temporal feature learning over 2D-CNNs. A 3D-CNN followed by a linear classifier was designed which outperformed other LSTM and pre-trained 2D-CNN based methods. Other studies include the integration of 3D-CNN and RNN networks Yao et al. [8] and convolutional two stream fusion network Feichtenhofer et al. [9]. These networks work well to derive spatio-temporal representations in video analysis, however at the cost of high computation which affects its real-time application and deployment in broadcasted events especially in games with rapid motions such as badminton.

In this work we aim to address the aforementioned issue. It should also be noted that one of the drawbacks of CNNs and LSTMs used so far in action classification frameworks is vanishing and exploding gradients, which result due to arbitrary scaling after each consecutive layer. These contribute to the computational complexity of the said models. Neural Accumulator (Trask et al. [10]) or NAC, on the other hand, maintains a consistent scale of the input owing to its weight matrix consisting of simply 1, 0 and -1. NAC works as an affine transformation layer where the scale of the output remains consistent as a result of its weight matrix. This enables multiple NAC units to be stacked together without dealing with the drawbacks, as faced by CNNs or LSTMs. We aim to exploit this characteristic of NAC in our video action classification method.

The main contributions of this paper are two novel end-to-end trained NAC based frameworks for action classification in video analysis. The proposed models need not be trained on GPUs. They achieve high classification accuracy in strikingly minimal training and testing time. For the purpose of comparison three deep learning based methods viz. Denoising FCN Autoencoder, Temporal convolutional network and CNN-LSTM (LRCN) for stroke classification have been implemented. These models were required to be trained on GPUs since they take more than 2 h to train on a CPU. The proposed models perform better, in terms of classification accuracy, than the comparing methods, except the LRCN. As far as the computation time, our models always exhibit lower training and testing time, even when they are run on CPU while the comparing methods on

GPU. In other words, if all are run on either GPUs or CPUs, the contrast in time difference (i.e the superiority of the proposed models) will be more prominent.

Experiments have been performed using 5-fold cross validation for several test-train splits varying from 10% to 50% to verify the effectiveness of the proposed model. The rest of the paper is arranged as follows, a brief discussion on related work in sports video action classification is presented in Sect. 2, followed by the proposed NAC based architectures and comparing methods in Sects. 3 and 4, respectively. In Sects. 5 and 6 we discuss different experimental protocols (viz. Dataset used, its pre-processing and preparation) and results obtained after thorough investigation. Lastly the effectiveness of the proposed method and future scope is summarized in Sect. 7. The code and dataset for this work is available on GitHub[1].

(a) Singles Match Frame_1 (b) Doubles Match_1, Frame_1 (c) Doubles Match_2 Frame_1

Fig. 1. Frame instances for three different Youtube badminton matches in UIUC2 dataset

2 Related Works

Badminton is a fast paced sport, analyzing a players strokes and performance can be exploited to be of some benefit to the player. Chu and Situmeang [11] studied classification of player strategy based on pose and stroke information of the player. Ghosh et al. [12] discussed an end to end framework for analysis of badminton videos, performing object detection, action recognition and segmentation. Furthermore, Yoshikawa et al. [13] proposed an automatic serve scene detection method using no prior knowledge, without segmentation and were able to extract motion and posture features of players using shift-invariance information. Following this by employing linear regression they detected specific scenes from the extracted features, and achieved high precision and recall values. Chu and Situmeang [11] clustered the player strategies into two categories namely

[1] https://github.com/poojacos/NAC-LSTM.

offensive and defensive and performed stroke classification on badminton video after detecting the players and court.

Studying players strategies retrieved through several successful badminton video analysis tools could add another dimension to the game play and preparation for the players. In this study, three 2006 Badminton World Cup match videos from UIUC2 dataset have been used which are far-view, low resolution as they are captured using static camera. Of the 3 matches one is a singles match and the other two are doubles matches as shown in Fig. 1. We extract the player instances from the video frames using segmented masks. These instances are then annotated into six different action sequences namely forehand, backhand, lob, serve, smash and react. The same stroke sequences of top and bottom players are classified together as one action. Serve and react sequences were found to be the least and the most respectively. Multivariate recognition is performed on the annotated actions and accuracies are reported. Similarity in poses of the players among different actions and low spatial resolution made the classification task more challenging.

3 Proposed NAC Framework for Stroke Classification

Two Novel NAC based frameworks have been proposed in this section. First one explores the advantages of using a NAC unit for feature extraction (spatial features per frame) as opposed to using a computationally expensive Autoencoder or a Convolutional neural network model. Following this the extracted features are fed as input to the LSTM model for classification. The second approach utilizes NAC in a way such that it learns the pixel-wise temporal dependencies from input frames, replacing LSTM. Additionally a dense layer carries out the multivariate classification in this scheme.

3.1 Neural Accumulator (NAC)

The neural accumulator (NAC) is a neural network unit where the weight parameter (W) assigned by these units are 1, 0 or -1. Trask et al. [10] proposed a differentiable and continuous parameterization of weights easily trainable with gradient descent.

$$Weights = tanh(\hat{W}) \cdot \sigma(\hat{M}) \tag{1}$$

where $\hat{W}$ and $\hat{M}$ (1) are randomly initialized. tanh is a hyperbolic function whose values lie between -1 and 1, whereas a sigmoid function lies between 0 and 1, hence there dot product ranges in between values [−1,1] with bias towards -1, 0 and 1. Two NAC implementations are simple and complex that support the ability of simple linear operations such as subtraction, addition and complex numeric operations i.e division and multiplication respectively. These two implementations with a gated logic form the basis for NALU or Neural Arithmetic Logic Unit. For the scope of this study we explored the utility of NAC for spatial and temporal feature extraction towards badminton stroke classification.

3.2 NAC for Spatial Feature Extraction

In this framework the prepossessed data sequences each made to be 44 frames long, and of frame dimension 32×32 are given as input to a NAC unit and transformed frame wise spatially. Number of NAC units stacked together is the same as the number of pixels in a frame, which is 1024 (32 * 32) in this case, to get the same number of input and output units. Here the intuition is to transform the *(number of frames * number of pixels)* input to a sparse representation with only relevant pixels kept non zero using the NAC layer. This transformed entity is then given as an input to an LSTM layer with number of LSTM cells equal to the number of frames to learn the relations of these pixels across the temporal dimension. The NAC unit will thus have two weight matrices of shape *(number of pixels * number of NAC units)*. In this case number of units is equal to the number of pixels, thus the number of parameters in this layer is equal to *2 * (number of pixels)*^{2}. This framework is trained end-to-end.

The major contribution of the proposed architecture is that it eliminates the requirement of using a CNN or an autoencoder based model for feature extraction. NAC easily distinguishes between relevant and non relevant pixels such as background pixels. The weights learned allow maximum signal to pass from the player body and racquet pixel positions that are most deterministic for a stroke such as hand and leg stances. Further the number of computations to extract features by NAC is significantly lower than in the convolutional frameworks owing to reduced number of matrix operations performed. A single layer of NAC units is sufficient compared to multiple convolutional layers. Learning a NAC function over the sequences followed by a LSTM proves to be sufficient to classify the strokes with high accuracy in significantly lesser training time. Considering x_{nac} as the input set of images (each frame of a sequence) to the NAC, and A_{lstm} be the output of NAC to be fed to a LSTM layer (as a sequence of 44 frames),

$$A_{lstm} = \sigma(Weights \cdot x_{nac}) \tag{2}$$

Where $Weights$ is initialized randomly as given by (1), and σ is the activation function. Following this the output gate of Lstm unit with A_{lstm} (2) as input can be defined as,

$$O_{lstm} = \sigma(W_o A_{lstm} + h_{t-1} U_o) \tag{3}$$

Where h_t in (3) is the hidden state, W_o and U_o are randomly initialized weights for output gate. Similarly input gate, forget gate outputs and output of hidden state can be estimated using A_{lstm} as input. For training of the proposed network Adam optimizer with a learning rate 1e−4 is used to minimize the mean square error function. The weights are updated with respect to the gradients calculated by the optimizer after every 32 training samples using backpropagation-over-time algorithm. Table 1 shows the number of parameters and output shape details of the designed framework.

Table 1. Detailed layer configurations NAC-LSTM model

Layer	Type	Output shape	Parameters
in_1	Input layer	44×1024	–
nac_1	NAC unit	44×1024	2097152
lstm_1	LSTM layer	6	24744
tanh_1	Tanh activation	–	–

3.3 NAC for Temporal Feature Extraction

In this framework the NAC unit is trained to learn meaningful temporal transformations of the features across time from the input sequence of frames. The intuition is to determine the inter dependency of a pixel at a fixed position across all the frames of a sequence. The input is of dimension *(number of pixels * number of frames)* to the NAC layer that outputs a compressed representation of size *(number of pixels * k)*, where $k \leq$ *number of frames*. When number of NAC units (k) is chosen to be less than the number of frames, this model performs dimensionality reduction. It can be thought of as the video being compressed to smaller number of frames where each non zero pixel in this form contains information most deterministic for a stroke. The outputs of NAC layer are then given as an input to a dense layer with sigmoid as activation function, for stroke classification. This framework has about the same parameters as the previous one, however its training time and testing time is halved. This because the weight matrices for NAC layer are of size *(number of frames * k)*. Thus even for a small image of size 32 * 32 this reduction in the size of weight matrix significantly improves the computation time. Considering x_{nac} as the input set of images (each frame of a sequence) to the NAC identical as the NAC-LSTM framework, and A_{dense} be the output of NAC.

$$A_{dense} = \sigma(Weights \cdot (x_{nac}.T)) \tag{4}$$

where $Weights$ are initialized randomly as given by (1), and σ is the activation function. The output A_{dense} (4) is flattened and given to a dense layer with sigmoid activation. It classifies the input into different stroke actions as shown in (5).

$$O_{dense} = \sigma(W_o A_{dense}) \tag{5}$$

where O_{dense} gives the output of the dense layer and W_o being weights initialized with glorot uniform. The batch size is taken as 32 and the model is trained with the optimizer Adam with a learning rate of 1e−4, the following Table 2 gives the detailed network configurations.

Table 2. Detailed layer configurations NAC-Dense model

Layer	Type	Output shape	Parameters
in_1	Input layer	1024×44	–
nac_1	NAC unit	1024×44	3872
flatten_1	Flatten layer	45056	–
dense_1	Dense layer	6	270342
sigmoid_1	Sigmoid activation	–	–

4 Comparing Methods

4.1 Denoising Autoencoder Fed LSTM for Stroke Classification

In this study a Denoising Autoencoder fed LSTM model was implemented where initially the autoencoder model was trained using stochastic gradient descent through backpropagation to optimize the mean square error function. Weights were updated after every defined batch size of noisy or corrupted training samples (noise factor of 0.2) with the rate: 1e−3 times the partial derivative of error w.r.t the initial weight. The LSTM model was trained on the latent space features extracted from the denoising autoencoder, multivariate classification required a one-hot coding of all classes. Further 5-fold cross validation was carried out for 300 epochs with Adam optimizer (learning rate = 1e−4) to acquire the accuracy for each of the defined classes.

The Encoder unit for the designed autoencoder consists of four weight layers, each convolutional, with three 7×7 and one 5×5 size filters. In between convolution layers, a simple max pooling operation is employed with kernel dimension 2×2. The decoder model has eight weight layers, each convolutional, with kernel dimensions identical to the encoder in an attempt to reconstruct the input. In place of a maxpooling layer in encoder the decoder has an upsampling layer with filter dimension 2×2. For adding non-linearity, Relu activation for encoder unit and LeakyRelu for decoder unit has been used, to prevent back propagating gradients from vanishing or exploding often faced when using sigmoid activation. In LSTM model, we experimented with both a NAC and a sigmoid unit as the top layer for class prediction.

4.2 TCN for Stroke Classification

TCN described by Lea et al. [14], namely Dilated TCN was utilized for experimentation. The Dilated-TCN model is similar to the wavenet architecture, where a series of blocks are defined with several convolutional layers. For the ease of combining activations from different layers, the number of filters is kept same for these layers. Each layer has a set of dilated convolution with the rate parameters. The dilation rate increases for consecutive layers in a block. A residual connection combines the layer's input and convolution signal.

The performance of dilated TCN model is evaluated for multiple dilation rate settings varying from 1 to 64 (i.e. 1, 2, 4, 8, 16, 32, 64) and with different number of filters ranging from 64 to 256. Training is done using the Adadelta optimizer for 110 epochs. In this work instead of using an additional spatio-temporal feature extractor and using the TCN to learn on those features, the TCN network is used to perform both the tasks since the image size is small (32×32) and the idea is to allow the network to learn the pixel wise variation across the frames of sequence instead of learning the variation of features.

4.3 Long Term Recurrent Convolutional Network (LRCN) for Stroke Classification

LRCN (CNN-LSTM) combines a deep convolutional model for spatial feature extraction and a separate model for temporal feature synthesis (Donahue et al. [5]). In this framework a Time Distributed convolution Network is designed followed by a RNN unit for sequence prediction. The implemented model has 4 time distributed convolutional layers. The filter dimension is kept same for all the layers, set to 3×3 and the number of filters chosen with ablative experimentation are 32 and 64. A time distributed MaxPool unit is employed after every two convolutional layers in an effort to check the number of trainable parameters. Use of regularizers such as batch normalization and dropout ensure that the network does not overfit the training samples. Following this, a bidirectional LSTM layer and a fully connected layer classify the input sequence of frames into 6 classes. The network is trained using the Adadelta optimizer adhering to the five fold cross validation scheme with the number of epochs equal to 80. Similar test train protocol is followed to verify the efficacy of the results.

5 Experiments

This section first provides an overview about the experiments performed followed by the dataset used and the pre-processing techniques employed.

5.1 Overview

Experiments have been performed on the badminton matches in UIUC2 dataset which were taken from Youtube 2006 badminton world cup matches. Every player is cropped out of the frames and the corresponding stroke played is determined. Six different stroke sequences played each consisting of 44 frames have been annotated. Following data preparation, total number of sequences were 427, 5-fold cross-validation has been used for training all the models. The split between train and test data was set at different percentages from 10% to 50% for thorough evaluation. Table 3 shows the data split statistics. For comparison Dilated Temporal convolutional networks (TCN), Autoencoder-LSTM and Long term Recurrent convolutional networks (LRCN) have been implemented.

In addition, the models were trained on a i7 7700 processor with Nvidia GPU 1050Ti and the GPU Tesla K-80. The models were implemented with Keras libraries using python as the programming language and Google Colab for GPU (K-80). The training time of different proposed models were between half to two hours.

Table 3. Test data statistics

Test data (%)	# Frames in Test set	# Frames in Train set
10	43	384
20	86	341
30	129	298
40	171	256
50	214	213

5.2 Dataset

In the dataset there are video frames from three different matches consisting of one singles and two doubles-matches. The total number of frames for the one singles and two doubles-matches were 3071, 1647 and 3936 respectively. Since the frames are derived from a Youtube video, the resolution of the images is quite low. Low quality of these images restricted us from using well researched approaches for action classification such as pose or posture estimation. It also prevented us from exploring the advantage of analysing footwork of the players for the purpose of strategy prediction. Utilizing the segmentation masks and bounding boxes data present in the dataset, we extracted every player's segmented image for the given frames from three matches. Total number of unique players from UIUC2 dataset can be estimated to be ten given one singles match and two doubles matches.

For this study we required player instances for different strokes played in order to classify them into six different annotated actions. There were several challenges to prepare the required data set such as occlusion and irregular frame instances per stroke. Occlusion made it difficult to extract the bounding boxes of players separately, hence we had to avoid all instances where severe occlusion occurred. In singles match dataset most of the bounding boxes for top and bottom players were easily separable, however in a few cases the stroke played could not be discerned. Occlusion posed a bigger problem with the doubles matches dataset, as it not only occurred between the top and bottom players but also among the two top and two bottom players themselves. Examples of occlusion is given in Fig. 2.

Fig. 2. (a),(c) and (b),(d) display occlusion instances in bottom and top players respectively for doubles matches dataset, (e) and (f) shows occlusion instances in singles match.

5.3 Data Preparation and Pre-processing

After extracting the bounding boxes of the top and bottom players separately from above discussed dataset, we manually annotated the strokes into six categories referencing from the initial badminton video frames. Different strokes recognised were react, forehand, backhand, lob, serve and smash (Fig. 3). An additional label chosen was no-play referring to the instances when the players were neither reacting to a stroke played by the opponent nor playing any of the defined strokes. These instances thus have no effect on game play and were thus discarded for the purpose of stroke classification.

The second challenge faced while data preparation was that an unequal number of frames for each of the defined strokes could be extracted due to occlusion. In order to avoid adding complexity to our stroke classification framework we required same number of frames for all the sequences. This was achieved by augmenting the initial set of extracted sequences uniformly to constitute a fixed number of frames. Each stroke sequence for simplicity and uniformity were made to have exactly 44 frames each. The extracted stroke sequence frames were made uniform and then converted to gray scale and resized to 32×32 for NAC-LSTM, CNN-LSTM models and 80×80 for Autoencoder-LSTM model, to reduce the number of trainable parameters and avoid over fitting, while extracting useful features. The total number of stroke sequences per dataset obtained by adhering to the above discussed protocols are presented in the Table 4. Furthermore, the

stroke sequences have been sample-wise normalised along the mean using the equation.

Table 4. Stroke sequences per dataset

Stroke	Doubles match_1	Doubles match_2	Singles match_1	Total
Backhand	19	22	21	62
Forehand	4	18	35	57
Lob	16	12	41	69
React	16	67	113	196
Serve	1	1	5	7
Smash	2	16	18	36

$$X_{norm} = \frac{X - X_{mean}}{X_{std}} \tag{6}$$

where, X_{norm} (in (6)) is the normalised vector X_{mean} is the mean of the sample X_{std} is the standard deviation of the sample. Data normalization is a powerful pre-processing tool that subdues the overall impact of outliers on the generalization of the network.

Fig. 3. Instances of different strokes annotated, (a) backhand, (b) forehand, (c) lob, (d) react, (e) serve and (f) smash

6 Results and Discussion

The following section discusses the performance evaluation and comparisons of the proposed NAC based models with other deep learning based approaches.

6.1 Average Classification Accuracy

NAC based frameworks have been implemented on CPU, in an effort to develop an algorithm capable of real-time applications in comparison with other models executed on GPUs as they are computationally intensive. Table 5 shows the classification accuracies obtained from LRCN, TCN, Autoencoder-LSTM and NAC (CPU) based models for different test data splits. The autoencoder-LSTM models with NAC and sigmoid units performed poorly on the combined dataset and took substantially more time (an average of 2 h on GPU) to train than the other models. However with the singles-match videos consisting a total of 233 sequences the model with NAC and Sigmoid achieved an average classification accuracy (over all the test split scenarios) of 85.37 and 84.66 respectively. LRCN was able to achieve the highest average accuracy over the combined dataset, although it took much longer for training than other frameworks. NAC-Dense and NAC-LSTM performed equally in terms of the average accuracy although the average training time (Table 6) taken by the NAC-LSTM was almost double to that of NAC-Dense, this is due to the difference between the number of NAC units stacked together in both the frameworks which in turn affects the size of weight matrices. In NAC-LSTM a total of 1024 (32 * 32) units determined by the dimension of the input frames were stacked together compared to NAC-Dense in which only 44 (Number of frames in a sequence) units were stacked.

Table 5. Accuracy Values for the discussed models

Models	Test data split					
	10%	20%	30%	40%	50%	Average
Autoencoder + LSTM + NAC	77.70	77.70	75	77.70	78.00	77.22
Autoencoder + LSTM + SIG.	81.00	75.00	80.50	83.30	83.30	80.63
Dilated TCN	83.72	83.3	84.11	83.72	83.41	83.65
LRCN	89.53	89.15	89.15	87.72	87.46	**88.60**
NAC + LSTM (CPU)	87.6	86.05	85.53	85.19	84.66	85.80
NAC + Dense (CPU)	87.6	86.43	85.95	85.26	84.29	85.90

6.2 Computation Time

The following Table 6, shows the training time taken by different models when run on GPUs in comparison to the proposed NAC-Frameworks run on CPU.

The proposed models are superior to other comparing models in terms of computation time. In addition, on CPU the comparing models take 2 to 8 h to train, whereas the proposed model exhibit better training time even when compared with the other models GPU runtime. For determining the deployability of the proposed networks in real time application the testing time evaluation is necessary. The following Table 7 shows the testing time for different models for the same protocol of test data split. NAC-Dense gives the best test time of *0.0453* s averaged over all the data splits. NAC-LSTM with the average test time of *0.1314* s performs better than the LRCN model (with avg. test time of *0.8359* s).

Table 6. Training time (in seconds) for the discussed models

Models	Test data split					
	10%	20%	30%	40%	50%	Average
Dilated TCN	498	742	764	918	947	773
LRCN	2305	2119	1922	1730	1537	1922
NAC + LSTM (CPU)	706	673	638	582	508	621
NAC + Dense (CPU)	358	322	304	279	264	**305**

6.3 Performance Analysis

The performance evaluation graph Fig. 4 is a plot between the average training time and the average classification accuracy over all test-train split scenarios. It can be observed that Dilated-TCN and NAC-LSTM have comparable training time, although the accuracy of NAC based model is still higher. Additionally it is evident that the trade-off between time and average accuracy is best for NAC-dense framework. To further verify the classification ability of the proposed models, single stroke classification with NAC-Dense has been performed as shown in Table 8. Six classifiers each attuned to a single stroke has been trained following Similar 10% to 50% test-train split protocol. The performance of the model is computed as the average for all classifiers over all splits which comes out to be 88.10.

Table 7. Test time (in seconds) for the discussed models

Models	Test data split					
	10%	20%	30%	40%	50%	Average
Dilated TCN	0.0398	0.0991	0.1738	0.1245	0.1529	0.1180
LRCN	0.3246	0.5165	0.9025	1.1310	1.3051	0.8359
NAC + LSTM (CPU)	0.0670	0.1320	0.1980	0.2600	0.3290	0.1314
NAC + Dense (CPU)	0.0180	0.0270	0.0650	0.0505	0.0660	**0.0453**

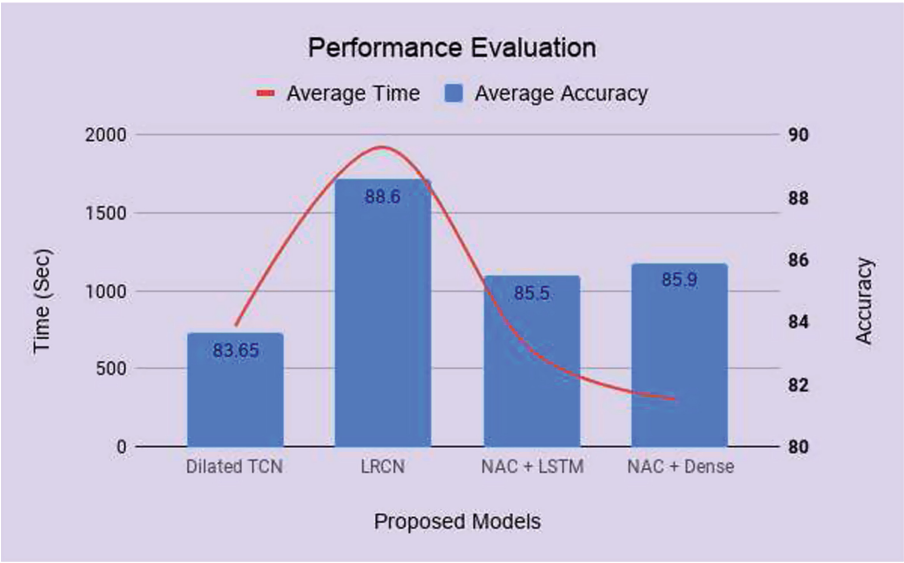


Fig. 4. Performance evaluation of the proposed Architectures

7 Conclusion and Future Work

In this work an attempt has been made towards developing an architecture to extract rich spatio-temporal features from videos in minimal time with less computational requirement. The ability of Neural Accumulator (NAC) to maintain a consistent scale enables it to preserve gradients over multiple stacked NAC units. This results in faster convergence of NAC based frameworks over other CNN and LSTM based models. Following this, two novel NAC based frameworks viz, NAC for learning spatial (NAC-LSTM) and temporal (NAC-Dense) transformations have been developed and applied for the task of action classification. as an example, in video analysis. The dataset used in this study is from UIUC2 which contains low resolution, far-view, static-camera videos of one 'singles' and two 'doubles' matches taken from Youtube. Six stroke sequences for all the three videos namely, forehand, backhand, lob, smash, serve and react have been annotated and pre-processed for the task of classification. Performance of the proposed models is compared with some state-of-the-art networks, e.g., FCN Autoencoder-LSTM, CNN-LSTM and Dilated TCN. 5-fold cross validation protocol for training, with test data splits from 10% to 50% have been used, and average accuracy values, training and testing time have been computed. The proposed models are superior to all the comparing models in terms of computation time. Performance (classification accuracy) wise they are also better than FCN autoencoder and Dilated TCN models. One may note that the superiority in computation time of the proposed models is achieved even when they are run on CPUs whereas the comparing models on GPUs. This contrast

becomes more prominent when all are run on CPUs or GPUs. In addition NAC-Dense achieves the best trade off between average classification accuracy and training time. Thus, for future work it can be employed for *real-time* application of broadcasted sport events with least computational requirement. Player and stroke-played dependencies could be learned via deep learning models and consequently player-wise stroke prediction in real time can be performed.

Table 8. NAC-Dense stroke wise classification accuracy

Models	Test data split					
	10%	20%	30%	40%	50%	Average
Backhand	83.72	87.21	87.21	85.96	84.58	85.736
Forehand	88.37	86.05	83.72	83.63	82.72	84.89
Lob	86.05	89.53	86.05	85.96	84.50	86.41
React	76.74	86.05	80.62	84.21	80.84	81.692
Serve	100	100	98.45	97.66	98.13	98.848
Smash	88.37	91.86	91.47	92.40	91.12	91.04
Overall average						88.10

Acknowledgment. The authors would like to thank Former Director and *INSA Distinguished Professor Sankar k. Pal*, Center for Soft Computing Research (CSCR), Indian Statistical Institute, Kolkata for providing guidance, facilities and his invaluable support.

References

1. Zhu, G., Xu, C., Huang, Q., Gao, W., Xing, L.: Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In: Proceedings of the 14th ACM International Conference on Multimedia, pp. 431–440. ACM (2006)
2. Zhao, Z., Elgammal, A.: Human activity recognition from frame's spatiotemporal representation. In: 19th International Conference on Pattern Recognition, pp. 1–4. IEEE (2008)
3. Pingali, G., Jean, Y., Opalach, A., Carlbom, I.: LucentVision: converting real world events into multimedia experiences. In: IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532), vol. 3, pp. 1433–1436. IEEE (2000)
4. Shah, M., Jain, R.: Motion-Based Recognition, vol. 9. Springer, Dordrecht (2013)
5. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)

6. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
7. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)
8. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A.: Describing videos by exploiting temporal structure. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4507–4515 (2015)
9. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
10. Trask, A., Hill, F., Reed, S.E., Rae, J., Dyer, C., Blunsom, P.: Neural arithmetic logic units. In: Advances in Neural Information Processing Systems, pp. 8035–8044 (2018)
11. Chu, W.T., Situmeang, S.: Badminton video analysis based on spatiotemporal and stroke features. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, pp. 448–451. ACM (2017)
12. Ghosh, A., Singh, S., Jawahar, C.: Towards structured analysis of broadcast badminton videos. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 296–304. IEEE (2018)
13. Yoshikawa, F., Kobayashi, T., Watanabe, K., Otsu, N.: Automated service scene detection for badminton game analysis using CHLAC and MRA. World Acad. Sci. Eng.Technol. **4**, 841–844 (2010)
14. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 156–165 (2017)

Fast GPU Convolution for CP-Decomposed Tensorial Neural Networks

Alexander Reustle, Tahseen Rabbani, and Furong Huang(✉)

Department of Computer Science, University of Maryland, College Park, USA
{areustle,furongh}@cs.umd.edu, trabbani@umd.edu

Abstract. We present a GPU algorithm for performing convolution with decomposed tensor products. We experimentally find up to **4.85x** faster execution times than Nvidia's cuDNN for some tensors. This is achieved by extending recent advances in compression of CNNs through use of tensor decomposition methods on weight tensors. Progress had previously been limited by a lack of fast operations to compute the decomposed variants of critical functions such as 2D convolution. We interpret this and other operations as a network of *compound convolution and tensor contraction on the decomposed factors* (i.e., *generalized tensor operations*). The prior approach sees such networks evaluated in a pairwise manner until the resulting output has been recovered, by composing functions in existing libraries such as cuDNN. The computational cost of such evaluations depends upon the order in which the index sums are evaluated, and varies between networks. The sequence of pairwise generalized tensor operations that minimizes the number of computations often produces large intermediate products, incurring performance bottlenecks when communicated with the scarce global memory of modern GPUs. Our solution is a GPU parallel algorithm which performs 2D convolution using filter tensors obtained through CP-decomposition with minimal memory overhead. We benchmark the run-time performance of our algorithm for common filter sizes in neural networks at multiple decomposition ranks. We compare ourselves against cuDNN traditional convolutions and find that our implementation is superior for lower ranks. We also propose a method for determining optimal sequences of pairwise tensor operations, achieving a minimal number of operations with memory constraints.

Keywords: Tensor methods · Neural network inference · Parallel algorithms

1 Introduction

Tensor decomposition methods have emerged as a means of training highly accurate convolutional neural networks while greatly reducing the number of model parameters [3,9,17,21,31]. So-called *Tensorial Neural Network* methods

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 468–487, 2021.
https://doi.org/10.1007/978-3-030-55180-3_35

involving the CANDECOMP/PARAFAC (CP) decomposition [16,31] in particular demonstrate significant promise; some models achieve accuracy scores 99% those of larger models with 1% of the parameters. These *Tensorial Neural Networks* express individual layers in deep neural networks as a graph of operations between input data and the factor tensors obtained from decomposing that layer's parameter tensor. This graph is referred to as a *tensorial neural network*, and these operations are *generalized tensor operations* [31]. Although these methods demonstrate significant promise, they are hampered by slow training speeds even on state-of-the-art hardware.

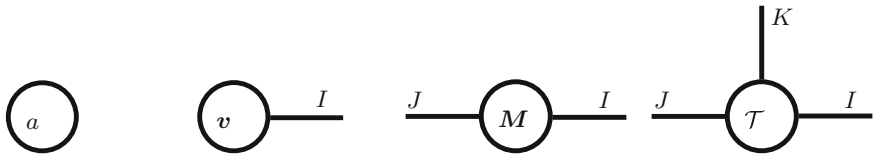
In this work we present our analysis of the causes of these slowdowns, along with part of our solution: a new GPU kernel to compute the forward pass of the critical 2D convolution using filter tensors derived from the CP decomposition. Written in CUDA, we refer to this as a *Fusion* method since it fuses the component operations of participating tensors, performing them in a single pass. This algorithm takes advantage of the many-thread concurrency of the GPU to execute the fused operation in parallel while minimizing communication with global memory. This algorithm does not achieve a minimal number of floating point operations; it redundantly recomputes intermediate values in parallel to avoid excessive global memory access. We benchmark our algorithm for runtime performance against the highly-optimized convolution algorithms in Nvidia's cuDNN library. As Fig. 6 shows our fusion algorithm is largely superior for a variety of common convolutional layer sizes found in neural networks [19].

We also propose an alternative approach, a graph search algorithm which determines the optimal sequence of pairwise operations needed to evaluate the *tensorial neural network*. This approach is a modification of a similar method in the field of quantum many-body physics: the `netcon` [28] solver for optimal sequences of *tensor network* contractions [5,7] where no convolution operations are involved. We extend this as `gnetcon` to support all *generalized tensor operations*, rigorously defined later in this paper.

The remainder of this work is organized as follows: the rest of Sect. 1 describes important background information on tensor networks, tensorial neural networks and the CP decomposition. Section 2 analyzes the 2D convolution operation in both the full-sized and CP decomposed domains. Section 3 describes our GPU algorithm for performing the convolution forward pass with decomposed filter factors. Section 4 presents our work on identifying optimal pairwise sequences for evaluating tensorial neural networks. In Sect. 5 we outline our testing and benchmarking methodology, and in Sect. 6 we present the results. We discuss related works in Sect. 7, and our conclusions in Sect. 8.

1.1 Generalized Tensor Operations

Following the convention in quantum physics [11,25], Fig. 1 introduces *tensor diagrams*, graphical representations for multi-dimensional mathematical objects. Here an array (scalar/vector/matrix/tensor) is represented as a *node* in the graph, and its *order* is denoted by the number of *edges* extending from the node,

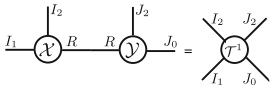


(a) Scalar **(b)** Vector **(c)** Matrix **(d)** Tensor

Fig. 1. Tensor Diagrams of a scalar $a \in \mathbb{R}$, a vector $\boldsymbol{v} \in \mathbb{R}^I$, a matrix $\boldsymbol{M} \in \mathbb{R}^{I \times J}$, and a 3-order tensor $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$.

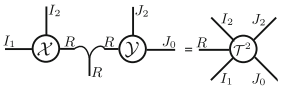
where each edge corresponds to one *mode* (whose *dimension* is denoted by the number associated to the edge).

An algebra of primitive tensor operations has been presented which when compounded generalize existing neural network architectures [31]. This extends the *tensor network* concept originally from the field of quantum and condensed matter physics [6] with operations that equate to higher-order multilinear evaluations of individual layers in a neural network, along with derivative and back-propagation rules.



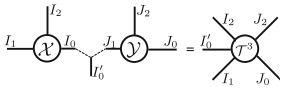
$$\sum_r \mathcal{X}_{r,i_1,i_2} \mathcal{Y}_{j_0,r,j_2} = \mathcal{T}^1_{i_1,i_2,j_0,j_2} \iff \mathcal{X} \times^R_R \mathcal{Y} = \mathcal{T}^1$$

(a) Mode-(R,R) tensor contraction.



$$\mathcal{X}_{r,i_1,i_2} \mathcal{Y}_{j_0,r,j_2} = \mathcal{T}^2_{r,i_1,i_2,j_0,j_2} \iff \mathcal{X} \otimes^R_R \mathcal{Y} = \mathcal{T}^2$$

(b) Mode-(R,R) tensor partial outer product.



$$\mathcal{X}_{:,i_1,i_2} * \mathcal{Y}_{j_0,:,j_2} = \mathcal{T}^3_{:,i_1,i_2,j_0,j_2} \iff \mathcal{X} *^{I_0}_{J_1} \mathcal{Y} = \mathcal{T}^3$$

(c) Mode-(I_0,J_1) tensor convolution .

Fig. 2. Primitives of generalized tensor operations. $\mathcal{X} \in \mathbb{R}^{I_0 \times I_1 \times I_2}$ and $\mathcal{Y} \in \mathbb{R}^{J_0 \times J_1 \times J_2}$ are input tensors, and $\mathcal{T}^1 \in \mathbb{R}^{I_1 \times I_2 \times J_0 \times J_2}$, $\mathcal{T}^2 \in \mathbb{R}^{I_0 \times I_1 \times I_2 \times J_0 \times J_2}$ and $\mathcal{T}^3 \in \mathbb{R}^{I'_0 \times I_1 \times I_2 \times J_0 \times J_2}$ are output tensors of corresponding operations. Existing tensor operations are only defined on lower-order $\mathcal{X}$ and $\mathcal{Y}$ such as matrices and vectors.

Three primitive operations are defined, the compound of which are *generalized tensor operations*. Figure 2 presents the primitives for *generalized tensor operations* on *high-order tensors*, extending the matrix/vector operations[1] using tensor diagrams. In tensor diagrams, an operation is represented by linking edges

[1] In Fig. 2, we illustrate these operations with simple examples of third-order tensors $\mathcal{X}$ and $\mathcal{Y}$, but they also apply for higher-order tensors as rigorously defined in [31].

from the input tensors, where the type of operation is denoted by the shape of line that connects the nodes: solid line stands for *tensor contraction/multiplication*, curved line is for *tensor partial outer product*, and dashed line represents *tensor convolution*. The *tensor contraction* generalizes matrix multiplication, while the *tensor partial outer product* generalizes the outer product for fibers of the operands. Finally, the *tensor convolution* can be defined by any convolution operation $*$ defined for 2 tensors.

A *generalized tensor operation* can be arbitrarily complicated, which can take more than two tensors as inputs, and multiple edges are linked simultaneously among the tensors (an example is Fig. 4b). In such a compound operation, different orders of evaluating the primitive operations yield the same result, though at the cost of different computational complexities. In general, it is NP-hard to obtain the best order to evaluate a compound operation with multiple tensor operands [20]. Using various tensor decomposition methods, Su et al. [31] convert the layers of existing network architectures into higher-order tensor network mappings, then use generalized operations to evaluate and retrain the nets. In doing so they demonstrate a significant reduction in model size while maintaining or in some cases improving upon the accuracy of the original network.

1.2 CANDECOMP/PARAFAC Decomposition

The CP decomposition [16] is a factorization of an order M tensor as a sum of R outer products between M vectors, where R is the tensor rank of the decomposition and each component vector's length is equal to the length of the corresponding mode in the original tensor. For example consider a 4-order tensor $\mathcal{K} \in \mathbb{R}^{T \times S \times H \times W}$, with component vectors $\boldsymbol{t}^{(r)} \in \mathbb{R}^{T}$, $\boldsymbol{s}^{(r)} \in \mathbb{R}^{S}$, $\boldsymbol{h}^{(r)} \in \mathbb{R}^{H}$, $\boldsymbol{w}^{(r)} \in \mathbb{R}^{W}$, the rank R decomposition of $\mathcal{K}$ is the sum (1).

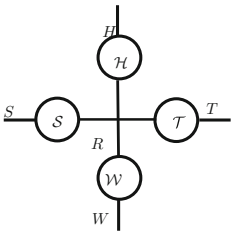


Fig. 3. Tensor diagram of CP decomposition of $\mathcal{K}$.

$$\mathcal{K} = \sum_{r}^{R} \boldsymbol{t}^{(r)} \otimes \boldsymbol{s}^{(r)} \otimes \boldsymbol{h}^{(r)} \otimes \boldsymbol{w}^{(r)} \tag{1}$$

where $\otimes$ denotes the vector outer product. It is common to concatenate the matching component vectors into M matrices of R columns. In our example this would produce four matrices $\mathcal{T} \in \mathbb{R}^{T \times R}, \mathcal{S} \in \mathbb{R}^{S \times R}, \mathcal{H} \in \mathbb{R}^{H \times R}, \mathcal{W} \in \mathbb{R}^{W \times R}$. We can now express the CP decomposition of $\mathcal{K}$ element-wise as (2).

$$\mathcal{K}_{tshw} = \sum_{t,s,h,w,r} \mathcal{T}_{tr} \cdot \mathcal{S}_{sr} \cdot \mathcal{H}_{hr} \cdot \mathcal{W}_{wr} \tag{2}$$

Using our tensor diagram notation described earlier, the CP decomposition of K would factorize the 4th-order tensor into four matrices, each sharing the R mode in a 4-way contraction, as demonstrated in Fig. 3 and in Eq. (3).

$$\mathcal{K} = \mathbf{1} \times_R^R (\mathcal{S} \otimes_R^R \mathcal{H} \otimes_R^R \mathcal{W} \otimes_R^R \mathcal{T}) \tag{3}$$

where $\mathbf{1}$ is an all-ones vector of length R.

1.3 Problem Description

The method of [31] has a drawback—the computational and memory cost of the existing implementation of the *generalized tensor operations* is high. This problem is mainly due to the following challenges:

1. Existing neural network frameworks like Tensorflow [10] and Pytorch [27] use tuned GPU library functions [4] to perform the critical operations of convolution and dense matrix multiplication. No such operations exist for tensor operations on decomposed kernels.
2. Consequently, researchers evaluate decomposed operations as a sequence of pairwise tensor operations composed of existing functions like `tf.nn.conv2d`. This introduces memory-access overhead due to materializing the intermediate products. In some cases the storage requirements for these intermediate products exceeds the available GPU memory.
3. Moreover, there is often no pre-existing implementation of the optimal pairwise sequence which minimizes the number of floating point operations when memory is constrained. Discovering such sequences is in general NP-hard.

Our **goal** is then to provide implementation schemes that are computation- and space- efficient. We propose two alternative solutions:

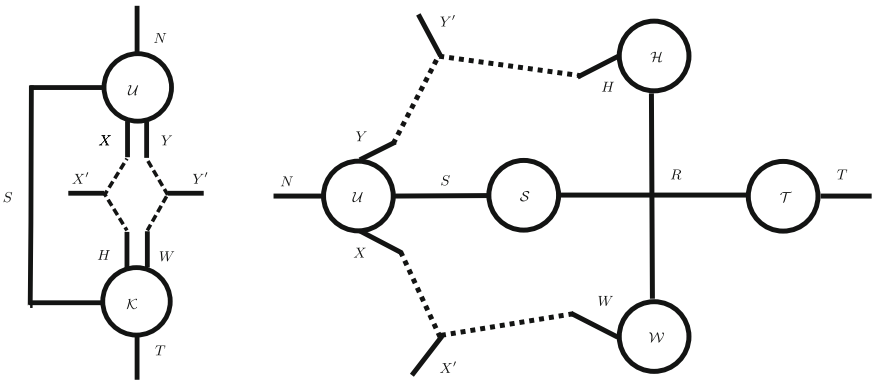
1. Fusing all the component operations in a *generalized tensor operation* into a memory minimal fused operation, avoiding computation and memory overhead/bottlenecks of the intermediate products.
2. Finding sequences for performing pairwise operations for any *generalized tensor operations* to achieve minimal number of floating point operations under some memory constraint.

1.4 Contributions

We introduce a *Fusion* method, a new GPU kernel to compute the forward pass of the critical 2-D convolution using filter tensors derived from the CP decomposition. We also propose an alternative approach, `gnetcon` a graph search algorithm which determines the optimal sequence of pairwise *generalized tensor operations* needed to evaluate the forward pass. Our fused approach implements a space/time trade-off. We store some small intermediate products to reduce onerous redundant computation, while redundantly computing other intermediates to maintain data locality and eliminate excessive global memory access. We confirm the speed of this approach empirically in Fig. 6 and Table 2. Similarly, we find that our fusion method significantly reduces total global memory usage as presented in Fig. 8 and Table 3. Further details provided in Sects. 5 and 6.

2 Convolution in Tensorial Neural Networks

The goal of our research is to take neural network layers which have been decomposed using the tensor decomposition methods (a.k.a., Tensorial Neural Networks (TNNs)) in [31] and execute them efficiently on a GPU in parallel. We



(a) $\mathcal{U}\left(*_H^Y \circ *_W^X \circ \times_S^S\right)\mathcal{K}$ **(b)** $\mathcal{U} \times_S^S \mathcal{S}(*_H^Y \circ \otimes_R^R)\mathcal{H}(*_W^X \circ \otimes_R^R)\mathcal{W} \times_T^T \mathcal{T}$

Fig. 4. Tensor diagram of **(a) Convolutional-layer in CNNs** and **(b) CP-decomposed Convolutional-layer in TNNs**. Both layers map a 4-order input tensor $\mathcal{U} \in \mathbb{R}^{N\times S\times X\times Y}$ to another 4-order output tensor $\mathcal{V} \in \mathbb{R}^{N\times T\times X'\times Y'}$, where X, Y and X', Y' are heights/widths of the input and output feature maps.

consider a layer in a TNN as a generalized neural network, that is a graph of *Generalized Tensor Operations* on the multiple component tensors within a layer. The component tensors consist of the input tensor and the weight tensors which have been decomposed from a higher-order convolutional layer.

2.1 Convolution-Layer in CNN

A traditional 2D-convolutional layer is parameterized by a 4-order kernel $\mathcal{K} \in \mathbb{R}^{H\times W\times S\times T}$, where H, W are height/width of the filters, and S, T are the numbers of input/output channels. Our implementation and experiments were conducted using the "channels first" data format for both the input tensor $\mathcal{U}$ and the output tensor $\mathcal{V}$. As illustrated in Fig. 4a, the operation is compound as in Eq. (4), where multiple primitive operations along different modes are executed. Specifically, two tensor convolutions at mode-(Y, H), mode-(X, W) and one tensor contraction at mode-(S, S) are performed simultaneously:

$$\mathcal{V} = \mathcal{U} \ \left(*_H^Y \circ *_W^X \circ \times_S^S\right) \ \mathcal{K} \tag{4}$$

Commonly convolution in neural networks is implemented as a cross-correlation [4], as we do here. The element-wise direct convolution of input tensor $\mathcal{U}$ with the filter tensor $\mathcal{K}$ is expressed in Eq. (5), although optimized variants such as Fast Fourier transform and Winograd convolution are often preferred for performance.

$$\mathcal{V}_{nty'x'} = \sum_{s,h,w} \mathcal{K}_{tshw} \cdot \mathcal{U}_{ns(y'+h)(x'+w)} \tag{5}$$

2.2 CP-Decomposed Convolution-Layer in TNN

A CP-decomposed Convolution-layer in tensorial neural network is parameterized by 4 decomposed kernels $\mathcal{S} \in \mathbb{R}^{S \times R}$, $\mathcal{H} \in \mathbb{R}^{H \times R}$, $\mathcal{W} \in \mathbb{R}^{W \times R}$ and $\mathcal{T} \in \mathbb{R}^{T \times R}$, as shown in Fig. 4b. The weight kernel $\mathcal{K}$ in Fig. 4a is CP-decomposed as

$$\mathcal{K} = \mathbf{1} \times_R^R (\mathcal{S} \otimes_R^R \mathcal{H} \otimes_R^R \mathcal{W} \otimes_R^R \mathcal{T}) \tag{6}$$

where $\mathbf{1}$ is an all-ones vector of length R, and H, W are height/width of the filters, and S, T are the numbers of input/output channels. Both tensor contraction $\times_R^R$ and tensor partial outer product $\otimes_R^R$ are primitives of generalized tensor operations as defined in Fig. 2.

TNN allows interactions between adjacent kernels through shared edges, crucial for modeling general multi-dimensional transformations without loss of expressive power. As illustrated in Fig. 4b, each mode of the input tensor $\mathcal{U}$ interacts with the one of the decomposed kernels. The forward pass is as follows: $\mathcal{U}_0 = \mathcal{U} \times_S^S \mathcal{S}$, $\mathcal{U}_1 = \mathcal{U}_0(*_H^Y \circ \otimes_R^R)\mathcal{H}$, $\mathcal{U}_2 = \mathcal{U}_1(*_W^X \circ \otimes_R^R)\mathcal{W}$ and $\mathcal{V} = \mathcal{U}_2 \times_T^T \mathcal{T}$, where $\mathcal{U}_0$, $\mathcal{U}_1$ and $\mathcal{U}_2$ are intermediate objects. We combine the above equations into a sequence of generalized tensor operation

$$\mathcal{V} = \mathcal{U} \times_S^S \mathcal{S}(*_H^Y \circ \otimes_R^R)\mathcal{H}(*_W^X \circ \otimes_R^R)\mathcal{W} \times_T^T \mathcal{T}. \tag{7}$$

Consider the naive single-element calculation for convolution between an order-4 input tensor $\mathcal{U}$, and four decomposed kernels $\mathcal{S}$, $\mathcal{H}$, $\mathcal{W}$ and $\mathcal{T}$

$$\mathcal{V}_{nty'x'} = \sum_{s,h,w,r} \mathcal{T}_{tr} \cdot \mathcal{S}_{sr} \cdot \mathcal{H}_{hr} \cdot \mathcal{W}_{wr} \cdot \mathcal{U}_{ns(y'+h)(x'+w)} \tag{8}$$

A single element in Eq. (8) minimally requires $(5SHWR)$ floating point operations to compute. However computing $\mathcal{V}_{n(t+1)y'x'}$ shares many common operations with $\mathcal{V}_{nty'x'}$ a fact generally true for $\mathcal{V}_{nt(y'+1)x'}$, $\mathcal{V}_{nty'(x'+1)}$ and other elements covered by the same filter. Computing the fiber of all T output channels using Eq. (8) takes $(5TSHWR)$ floating point operations. To compute $\mathcal{V}_{n:y'x'}$ while minimizing redundant computation for an entire fiber of T output channels, we must share intermediate products.

Applying the distributive property of scalar arithmetic, and storing the intermediate accumulation in a vector of length R, we can express (8) as (9):

$$\begin{aligned} \boldsymbol{a}_r &= \sum_{s,h,w} \mathcal{S}_{sr} \cdot \mathcal{H}_{hr} \cdot \mathcal{W}_{wr} \cdot \mathcal{U}_{ns(y'+h)(x'+w)} \\ \mathcal{V}_{nty'x'} &= \sum_r \mathcal{T}_{tr} \cdot \boldsymbol{a}_r \end{aligned} \tag{9}$$

Computing (9) would require $(4RSHW + 2TR)$ floating point operations for a fiber of T elements. This is more efficient when $T < 1/2SHW$, a fact commonly true in neural network layers. With the added trade off of storing the intermediate accumulation vector. Similar intermediate storage options are available for the other modes of the tensor.

3 GPU Fused CP Convolution Operation

All results presented in this paper report performance obtained on the Nvidia RTX 2080-Ti (Turing) GPU. This device has a theoretical maximum single precision (32-bit) floating point performance of 14.1 TFLOPS, which it achieves with 4352 CUDA cores spread across 68 streaming multiprocessors on chip [8]. This parallelism is exposed to the programmer in an organizational hierarchy of computation with *threads*, *warps*, *blocks* and *grids*.

Threads are the smallest unit of compute and are organized into *warps* of at most 32 sequential threads executing simultaneously, and further grouped into *blocks* of execution that may communicate via shared memory. A *grid* of blocks is executes in an arbitrary sequence to perform the desired calculation on data resident in the devices global memory. The 2080-Ti has theoretical peak bandwidth for global memory access of 616 GB/s. While impressive this is insufficient to promptly deliver data to the all active threads on chip. Thus many compute kernels are performance limited by memory load/store operations, not FLOP count. Such kernels are known as *bandwidth-bound* kernels. The 2080-Ti alleviates this bottleneck somewhat with a small L1/L2 cache, which can improve the performance of *bandwidth-bound* kernels that exploit data locality. The GPU also exposes a set shared memory that may be accessed collaboratively by threads in a block, a core method of thread communication. The other is the use of warp-level primitives, where threads in a warp communicate directly.

3.1 Naive Implementation

Algorithm 1. Naive CP Convolution single element

Input: n, t, x', y'

for $r < R$ **do**
  for $s < S$ **do**
    for $h < H$ **do**
      for $w < W$ **do**
        $\mathcal{V}_{n,t,y',x'} \mathrel{+}= \mathcal{T}_{tr}\mathcal{S}_{sr}\mathcal{H}_{hr}\mathcal{W}_{wr}\mathcal{U}_{n,s(y'+h)(x'+w)}$
      end for
    end for
  end for
end for

A naive implementation of Eq. (8) can be seen in Algorithm 1. The deeply nested loops redundantly recompute many subproduct terms, and share neither intermediate products, nor common input elements. This latter part causes it to suffer from substantial bandwidth constraints as adjacent threads load repeated data elements from global memory.

3.2 Proposed Implementation

For our algorithm to utilize the parallelism provided by the GPU we must split the input tensor into small independent tiles of data. The output elements obtained from these tiles are computed at the block level, independently of other blocks in the grid. Thus we introduce a redundant computation trade off: intermediate results obtained from data elements fully in the block are shared, while those in the in boundary regions must be recomputed in each tile.

The fused kernel accepts input data from tensor $\mathcal{U}$ in the "channels first" data format, the format most favorable to cuDNN. Thus a 4-order tensor $\mathcal{U}$ has modes batch size (N), input channels (S), feature height (Y), and feature width (X) ordered from least to most frequently varying.

In our Algorithm 2, a block of threads is allocated for each tile of input. At kernel launch, the threads are assigned coordinates denoting which y', x' input they will convolve, and which subset of s channels they will contract. As input channels often outnumber threads available for contraction, threads loop over channels, striding by the channel tile size "S_TILE". Each thread maintains a local vector of length R into which intermediate values accumulate. Assuming small R this vector can be maintained entirely in a thread's local registers. Threads read input data from a few contiguous regions of global memory, to maximally utilize global memory bandwidth.

Algorithm 2. GPU Fused CP Convolution

Input: $n, t, x', y', \mathcal{U}, \mathcal{T}, \mathcal{S}, \mathcal{H}, \mathcal{W}$

{Allocate Length R vectors for local intermediates}

$\boldsymbol{q} \leftarrow 0$
$\boldsymbol{p} \leftarrow 0$
for $s <$S ; stride == S_TILE **do**
  $\boldsymbol{p} \leftarrow 0$
  for $h < H$ **do**
    for $w < W$ **do**
      $\boldsymbol{p} \leftarrow \boldsymbol{p} + \mathcal{U}_{n,s(y'+h)(x'+w)} \circ \mathcal{H}_{h:} \circ \mathcal{W}_{w:}$
    end for
  end for
  $\boldsymbol{q} \leftarrow \boldsymbol{q} + \boldsymbol{p} \circ \mathcal{S}_{s:}$
end for
for $t < T$ **do**
  $a \leftarrow \langle \boldsymbol{q}, \mathcal{T}_{t:} \rangle$
  $a \leftarrow$ `WarpReduce`(a, S_TILE_INDEX)
  `SyncWarp`
  if S_TILE_INDEX== 0 **then**
    $\mathcal{V}_{nty'x'} \leftarrow a$
  end if
end for

Algorithm 2 assumes the tensor dimensions and the memory constraints are known at runtime, which is satisfied in TNNs. It further assumes that minimizing FLOPs count within the provided memory constraint is sufficient for finding an optimal sequence. If gnetcon fails to satisfy the memory constraint, another approach (such as fusion) with a smaller memory footprint must be used.

Most global data loads occur in the inner-most loop at line 7. Warps load contiguous chunks of $\mathcal{U}$ and perform an element-wise Hadamard product on the length R row vectors of the filter factors $\mathcal{H}$ and $\mathcal{W}$. Here we exploit the data locality of the L1 and L2 cache. The participating elements from $\mathcal{U}$ will be shared by adjacent warps. Sequential warps which depend upon that region of memory are very likely to find it in the cache when executing, avoiding global memory loads. Warps also share the same row of the filter tensors $\mathcal{H}$ & $\mathcal{W}$, which is also cached.

The vector $\boldsymbol{p}$ is local to the thread and never shared. It is element-wise scaled with a slice of the input channel tensor $\mathcal{S}$ for each thread at line 10. Input channel sizes are commonly large in neural network layers. Accessing that chunk of global memory is likely to eject the previously cached loads and cause a reduction in performance, so we perform it independently of the operation at line 7.

After striding over the input channels of the tensor, each thread now contains a length R vector $\boldsymbol{q}$, which is a partial sum. Looping over output channels $t \in T$, at line 13 each thread performs an inner product calculation with its local $\boldsymbol{q}$ and the output channel tensor for t, accumulating the product in the scalar a. These threads are adjacent, thus executing in the same warp, thus at line 14 we use the warp level primitives to share intermediates, performing a reduction on the values in a spread across each thread. This performs the reduction in at most $\log_2(32) = 5$ operations, and accumulates the final sum in the a variable of the warp with index 0. Line 15 is a barrier for the warp of threads ensuring they have all completed, thus the accumulated sum is correct. This synchronization barrier does not extend to the rest of the block, and so does not hinder performance. Finally the thread which has accumulated the result writes it out to global memory in the output tensor $\mathcal{V}$.

4 Optimal Operation Sequences

Minimizing the number of floating point operations in the evaluation of an input tensor is a standard approach we consider and measure against the fusion approach. To develop an algorithm which could determine the minimal number of operations needed to evaluate an input in a tensorial representation of a neural network, we extend the techniques used in networks consisting solely of contractions, which are very well-studied. Consider the following sequence of tensors,

$$\sum_{i,j,k,l} \mathcal{X}_{ijk}\mathcal{Y}_{ikl}\mathcal{Z}_{mln} + \sum_{p,q} \mathcal{W}_{pr}\mathcal{V}_{qs}, \tag{10}$$

where the sums are over same-dimensional modes between tensors, i.e., contractions. A classical question arises when attempting to evaluate such a network: should memory consumption or the number of intermediary operations be minimized? In this section, we concern ourselves with the latter. For a simpler example, consider a matrix multiplication such as $\boldsymbol{A} = \boldsymbol{B} \times \boldsymbol{C} \times \boldsymbol{D}$ which consists

of a sequence of contractions $\boldsymbol{A}_{ij} = \sum_k \boldsymbol{B}_{ik}\boldsymbol{C}_{kl}\boldsymbol{D}_{lj}$. One may ask if $(\boldsymbol{B}\boldsymbol{C})\boldsymbol{D}$ or $\boldsymbol{B}(\boldsymbol{C}\boldsymbol{D})$ costs less floating-point operations. Efficient contraction of tensor networks has a vast literature, and appears in many quantum computational chemistry problems. The problem rapidly becomes intractable if the network contains many tensors with a total collection of many modes.

Finding the optimal contraction sequence which minimizes the number of floating point operations is known to be NP-hard, but fast algorithms do exist. One such well-known breadth-first algorithm is `netcon()` due to Pfeifer et al. [28]. We refer to the summary of the breadth-first approach as described in [28]:

1. Let $L_1 = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ be the set of tensors in the network.
2. Let i be an index counter from 2 to n. For each i:
   (a) Let L_c be the set of all possible subnetworks created by contracting i tensors from L_1.
   (b) For each pair of sets L_d, L_{c-d}, $1 \leq d \leq \left\lfloor \frac{c}{2} \right\rfloor$, and for each $\mathcal{T}_a \in L_d$, $\mathcal{T}_b \in L_{c-d}$ such that each element of L_1 appears at most once in the subnetwork $(\mathcal{T}_a\mathcal{T}_b)$:
      i. Compute the cost c of contracting $\mathcal{T}_a$ with $\mathcal{T}_b$.
      ii. If $\mathcal{T}_a$ and/or $\mathcal{T}_b$ are not in L_1, then add the cost of constructing of $\mathcal{T}_a$ and/or $\mathcal{T}_b$ to c.
      iii. Let the contraction sequence $\mathcal{S}$ for constructing this subnetwork be written $S = (\mathcal{T}_a\mathcal{T}_b)$. If $\mathcal{T}_a$ and/or $\mathcal{T}_b$ are not in L_1, then optimal contraction sequences for $\mathcal{T}_a$ and $\mathcal{T}_b$ will have been recorded already. In $\mathcal{S}$, replace the occurrences of $\mathcal{T}_a$ and/or $\mathcal{T}_b$ with these sequences.
      iv. Locate the subnetwork in L_c which corresponds to $(\mathcal{T}_a\mathcal{T}_b)$. If c is the cheapest cost for constructing this subnetwork, record c and the associated contraction sequence $\mathcal{S}$ against this subnetwork.
3. The optimal cost c_{best} and associated sequence $\mathcal{S}_{\text{best}}$ are recorded against the only element of L_n.

`netcon()` contains a cost-capping feature: subnetworks may be rejected for operation if the intermediate product exceeds a predefined memory constraint and other cheaper paths to the final product are available. While `netcon()` may be used to evaluate a network such as Fig. 5a, it cannot be used to evaluate the generalized tensor operations which involve, for instance, convolutions and partial outer products. For instance, one layer of the network $\mathcal{A} *_7^3 \mathcal{B} \times_6^6 \mathcal{C} \otimes_4^4 \mathcal{D}$ in Fig. 5b.

One of the core contributions of this paper is a generalization of `netcon()` which we refer to as `gnetcon()` capable of finding the optimal operation sequence for a given tensorial neural network. `gnetcon()` modifies `netcon()` by introducing an updated cost model to handle any generalized operation. For $\mathcal{U} \in \mathbb{R}^{I_0 \times I_1 \times \ldots \times I_{m-1}}$, $\mathcal{V} \in \mathbb{R}^{J_0 \times J_1 \times \ldots \times J_{n-1}}$, we introduce the following floating point operation complexities[2] to obtain optimal pairwise sequence for the generalized tensor operations:

[2] Note that for convolution cost (12), we assume no Fast-Fourier Transform is used.

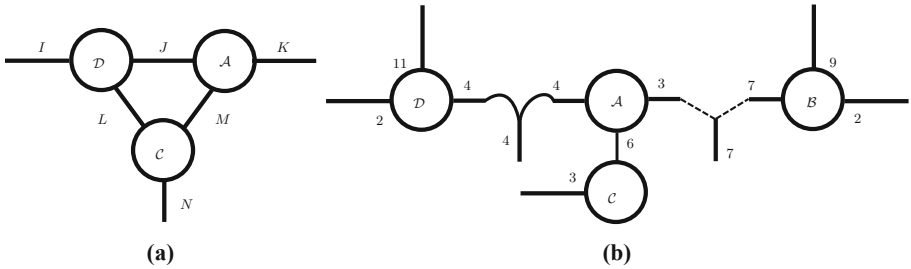


Fig. 5. Example networks. **(a)** An example of tensor network. **(b)** An example of 1 layer of a deep tensorial neural network involving generalized operations.

$$\text{cost}[\mathcal{U} \times_l^k \mathcal{V}] = O((\prod_{u=0}^{m-1} I_u)(\prod_{v=0,v\neq l}^{n-1} J_v)) \tag{11}$$

$$\text{cost}[\mathcal{U} *_l^k \mathcal{V}] = O((\prod_{u=0}^{m-1} I_u)(\prod_{v=0}^{n-1} J_v)) \tag{12}$$

$$\text{cost}[\mathcal{U} \otimes_l^k \mathcal{V}] = O((\prod_{u=0}^{m-1} I_u)(\prod_{v=0,v\neq l}^{n-1} J_v)). \tag{13}$$

Furthermore, `gnetcon()` maintains the cost-capping feature of `necton()` and thus can handle predefined memory constraints. As an example, running `gnetcon`$(\mathcal{A} *_7^3 \mathcal{B} \times_6^6 \mathcal{C} \otimes_4^4 \mathcal{D})$ for the generalized tensor operation in Fig. 5b, we obtain the optimal pairwise operation sequence $((\mathcal{A} \times_6^6 \mathcal{C}) *_7^3 \mathcal{B})) \otimes_4^4 \mathcal{D})$. The above costs 38,016 floating point operations, whereas a naive implementation in a order such as, $(((\mathcal{A} *_7^3 \mathcal{B}) \times_6^6 \mathcal{C}) \otimes_4^4 \mathcal{D})$, costs 45,360 floating point operations.

Using `gnetcon()`, we can execute a forward pass in a tensorial neural network according to an operation-minimizing strategy.

Forward Passes in CP-decomposed Convolution-layer in TNN As Generalized Tensor Operation Sequences. In a CP-convolutional layer as shown in Fig. 4b, the forward pass is a general tensor operation sequence $\mathcal{V} = \mathcal{U} \times_S^S \mathcal{S}(*_H^Y \circ \otimes_R^R)\mathcal{H}(*_W^X \circ \otimes_R^R)\mathcal{W} \times_T^T \mathcal{T}$. Once the dimensions of $\mathcal{U}$, $\mathcal{S}$,$\mathcal{T}$,$\mathcal{H}$ and $\mathcal{W}$ are set, Eq. (7) is passed into `gnetcon()` to determine the optimal order of operations. Similarly, our`gnetcon()` determines the minimal number of floating operations needed to carry out a forward pass in one layer of a convolutional neural network (CNN). We measure the time needed to execute a forward pass in the order recommended by `gnetcon()` and use these times as a baseline comparison against the fusion approach.

5 Performance Benchmarking

To test the correctness and performance of our CP Convolution kernel we relied heavily on facilities provided by the CUDA library. All tests were conducted on a private workstation running 64-bit Ubuntu 16.04, with a 12 core Intel Xeon CPU E5–1650 v4 CPU and 64 GB RAM. The GPU is an Nvidia RTX 2080-Ti

with driver Version: 418.87.00. Our implementation was compiled using CUDA Version: 10.1.243, while cuDNN version 7.4.2.24 was used as both a benchmark and an oracle for operator correctness.

5.1 Correctness Testing

All output values from the fused CP convolution kernel were verified as correct to a floating point tolerance of $\epsilon = 10^{-5}$. The testing procedure was as follows. In advance, we determined a list of input and filter tensor shapes. For all shapes in the list we allocated in GPU global memory a 4th-order input tensor $\mathcal{U}$ with uniformly random elements between 0 and 1. These elements were generated by cuRAND using a constant seed of "1234". We also generated four filter matrices with S, T, Y, and X rows respectively, and R columns, where R is the rank of the CP decomposition. We vary R for the tests and benchmarks, allowing it to take on the values $[1, 2, 4, 8, 16]$.

The 4-order, rank R filter tensor $\mathcal{K}$ was composed from the CP factor matrices by applying Eq. (1). The traditional convolution $\mathcal{V} = \mathcal{U} * \mathcal{K}$ was computed using the function *cudnnConvolutionForward*, and a cuDNN convolution algorithm gotten with the `CUDNN_CONVOLUTION_FWD_PREFER_FASTEST` selection. The CP convolution was calculated using our algorithm to produce $\mathcal{V}$. Implicit padding values were chosen to replicate a "SAME" padding often used in deep neural networks.

Correctness was determined by comparing each element of tensors $\mathcal{V}$, and $\mathcal{V}'$ for approximate equality to within a floating point tolerance of $\epsilon = 10^{-5}$. This was done using both a CPU library function and a custom GPU comparison function. The CPU library was the C++ *DocTest* unit test framework. The custom GPU comparison implemented "close enough" comparison from [14] §4.2.2, Eq. 37. Both were used for verifying the final kernel.

5.2 Fused CP Convolution Performance Benchmarking

Input tensors for timing benchmarks were generated using the same method used for correctness testing. The sizes and shapes of these tensors are described in more detail in Table 1. All are stored on device in GPU global memory before benchmarking began. The cuDNN library provides many algorithm options for forward convolution in the channels first data format. We executed the one selected by cuDNN using the `CUDNN_CONVOLUTION_FWD_PREFER_FASTEST` algorithm preference. Many of these algorithms require some amount of "workspace memory" to be allocated in global memory, which was fully provided for our tests.

Timing values were captured by decorating the kernel launch code with profiling calls using the `cudaEventRecord` features exposed by the CUDA API. Calls for start and stop to `cudaEventRecord` were placed immediately before the kernel launch, and after the `cudaSynchronizeDevice` call to ensure full recording of only the kernel launch and complete execution. Each operation was repeated 47 times and the duration of each run was captured. Results presented are the arithmetic mean of all run durations expressed in microseconds (μs).

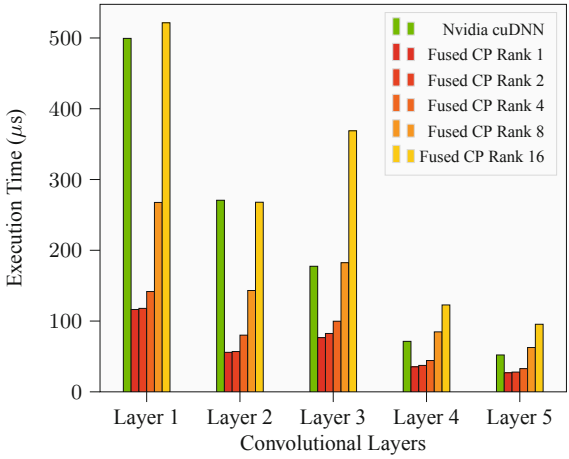


Fig. 6. Fused CP Convolution Execution Time of different operators on common convolution layers of neural networks Nvidia cuDNN with full-sized filter tensors vs. our Fused CP Convolution operation at various decomposed tensor ranks.

5.3 Pairwise Sequential Convolution Performance Benchmarking

We also benchmark the performance of our optimal pairwise sequence of *generalized tensor operations* using the TensorFlow deep neural network framework [10]. We express the graph of tensors in tensor diagram and generate the order to evaluate shared edges. Results are obtained using the built-in `tf.test.Benchmark` class and Tensorflow version 1.14.

Table 1. Convolution Operation Data Sizes. Tensor sizes taken from common convolution layers in neural networks. All batch sizes are 1. (S): # of input channels. (Y): feature tensor height. (X): feature tensor width. (T): # of output channels. (H): filter height. (W): filter width. (Rank) $\in \{1, 2, 4, 8, 16\}$.

Convolution layer	(S Y X)	(T H W)	# Features	# Filter params.	# CP Filter params.
1	(3, 224, 224)	(96, 11, 11)	150,528	34,828	121·Rank
2	(48, 55, 55)	(256, 5, 5)	145,200	307,200	314·Rank
3	(256, 27, 27)	(384, 3, 3)	186,624	884,736	646·Rank
4	(192, 13, 13)	(384, 3, 3)	32,448	663,552	582·Rank
5	(192, 13, 13)	(256, 3, 3)	32,448	442,368	454·Rank

6 Benchmarking Results

Our benchmark results are summarized in Fig. 6 and Table 2. As Fig. 6 shows our fusion algorithm is superior to cuDNN in most low-rank instances for common convolutional layers. Fused CP convolution is the faster in most cases below rank 16.

The fastest relative performance occurred at the rank 1 decomposition of layer 2. **Our code performed 4.85x faster than cuDNN with 75% of the memory usage**. The superior performance of our Fused CP convolution kernel extends even to higher ranks. The rank 4 run of deep layer 4 ran **1.61x faster, while using 26.1x less memory**.

A deeper look suggests that our algorithm scales linearly with rank, note the rank step size increments in sequential powers of 2. Another observation is that all ranks of fused CP convolution scale very well with channel depth, but somewhat poorly with input feature height and width. This is a limitation shared by cuDNN for traditional convolution.

Table 2. Execution Time in μs. cuDNN convolution benchmarked against our Fused CP-convolution operator, and our Pairwise optimal sequencer implemented in TensorFlow. Data sizes defined in Table 1.

Layer:	cuDNN (μs)	Fused CP convolution (μs)					Pairwise sequential CP convolution (μs)				
		Rank 1	2	4	8	16	Rank 1	2	4	8	16
1	499.4	**116.3**	**117.9**	**141.7**	**267.6**	521.5	3069.5	3178.9	3308.4	3519.5	3720.0
2	270.7	**55.8**	**56.9**	**80.1**	**143.2**	**267.9**	2079.7	1911.0	1838.0	1961.8	2073.7
3	177.4	**76.6**	**82.5**	**99.8**	182.5	368.8	2033.9	2047.5	2066.2	1985.0	1936.4
4	71.3	**35.4**	**37.3**	**44.2**	84.7	122.8	1810.0	1812.5	1907.5	1860.3	1759.7
5	52.1	**26.9**	**27.9**	**32.8**	62.6	95.5	1741.4	1790.6	1779.8	2399.3	1761.5

The pairwise sequential operation implemented in TensorFlow does not share the same superior performance characteristics as our fused convolution kernel. This is starkly visible in Fig. 7 which plots the execution time of the pairwise sequential forward convolution for various ranks of the different convolutional layers. We contribute much of this to the overhead introduced by tensorflow, which will automatically manage migration of intermediate tensors in and out of device memory during execution. Nevertheless this comparison is warranted as TensorFlow, along with the other popular neural network frameworks, remains the only widely available means of evaluating a *tensorial neural network* layer.

Our pairwise sequencer, implemented in TensorFlow, is the current state-of-the-art for TNN implementations, which are not currently addressed by other libraries like cuDNN. It is entirely possible that we registered slower execution times exclusively due to TensorFlow overhead. The cuDNN "baseline" we compare our pairwise sequencer against is an "equivalent" convolution after merging the sequence of operations being processed by our pairwise sequencer into one convolution; therefore not a fair comparison. A fair comparison would be to implement the sequence directly in cudnn library calls, a non-trivial task that we defer to a future work.

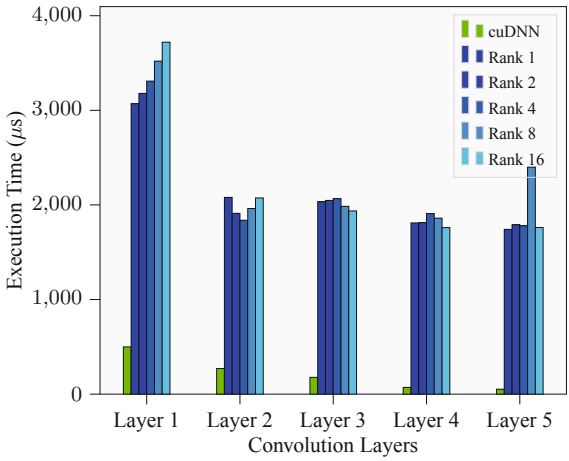


Fig. 7. Pairwise Sequential Forward Convolution Execution Time of different operators on common convolution layers of neural networks Nvidia cuDNN with full-sized filter tensors vs. our pairwise sequential CP Convolution operation at various decomposed tensor ranks.

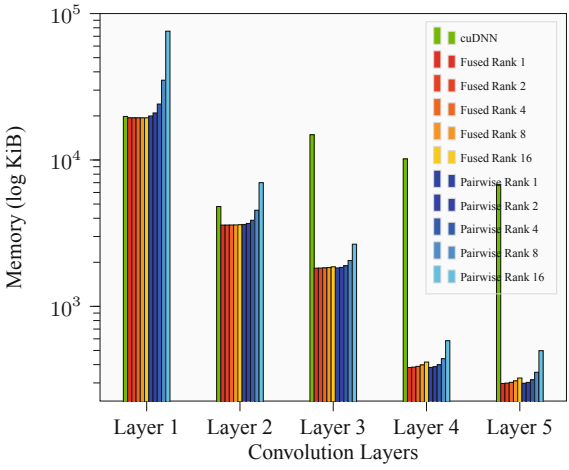


Fig. 8. Global Memory Usage (log KiB) of different operators on common convolution layers of neural networks, including traditional cuDNN convolution, Fused CP convolution, and pairwise sequential CP convolution at various decomposed tensor ranks.

Turning to the memory usage in Fig. 8 and Table 3, we see that the fused operator uses the least memory in all cases. We use a log-scale along the vertical axis for better visualization due to the scale difference between layers. All operations materialize the input and output feature tensors, which account for

the bulk of the memory footprint in the shallower layers. The increased cuDNN memory usage in later layers is largely a consequence of the extra "workspace memory" these particular cuDNN algorithms require. The values for the pairwise sequential convolution memory usage express the cumulative total of all participating tensors, including intermediate products. The full impact of the large footprint is alleviated somewhat by TensorFlow, which will act to stream intermediate data out of GPU memory and into host memory to reduce global memory pressure. Thus the true allocation size maximally resident in the GPU at during pairwise sequencing is often lower than the cumulative values.

Our two approaches are state-of-the-art for TNNs. There are directions to improve our approaches such as FFT, Winograd, GEMM lowering, or Tensor Core approaches. Such implementations represent possible future work.

Table 3. Global Memory Usage (KiB) of different operators on common convolutional layers of neural networks, including traditional cuDNN convolution, Fused CP convolution, and pairwise sequential CP convolution at various decomposed tensor ranks.

Layer:	cuDNN (KiB)	Fused CP Convolution (KiB)					Pairwise Sequential CP Conv. (KiB)				
		Rank 1	2	4	8	16	Rank 1	2	4	8	16
1	19834	**19405**	**19405**	**19406**	**19408**	**19412**	19992	20972	24108	35085	75854
2	4810	**3593**	**3595**	**3597**	**3602**	**3612**	**3628**	**3687**	**3877**	4539	6999
3	14880	**1825**	**1828**	**1833**	**1843**	**1863**	**1832**	**1847**	**1895**	**2059**	**2659**
4	10174	**383**	**385**	**389**	**398**	**417**	**383**	**387**	**399**	**439**	**583**
5	6825	**298**	**299**	**301**	**310**	**324**	**299**	**303**	**315**	**355**	**498**

7 Related Works

In the field of unsupervised learning, advancement has come through applying tensor decomposition methods to the problem of learning latent variable models [1]. Much research on tensor decompositions is directed at approximations of the CP decomposition described in Sect. 1.2; leading to research applying tensors and tensor decompositions to neural networks. In [18], the authors replace fully connected layers and the flattening step necessary to transition to them from convolutional layers with a novel pair of tensor-based layers, retaining structural information. Another instance of tensorizing existing neural network layers appears in [21] where the authors reduce the processing time incurred on convolution kernels through sequential application of CP decomposed factor tensors. This early work demonstrated the potential for considerable speedups in CPU implementations of convolution using tensor decomposition.

Ideas for fully tensorizing neural networks are also popular. The authors in [24] transform the dense matrix weights of fully connected layers into decomposed tensors represented with the Tensor-train decomposition from [26]. This can be applied to all fully connected layers in a network, resulting in a totally

tensorized neural network. Tensor decomposition methods were used to prove theoretical guarantees on the generalization error of two-layer neural networks in [12], and deep CP decomposed CNNs by [22], with state of the art generalization guarantees proven by statistical bounds on generalization error.

Other fields have focused on efficient tensor contraction due to the primacy of contraction in molecular chemistry models. The Tensor Contraction Engine automatically compiles contraction code for high performance computing environments [2]. This was extended to GPU code in [23].

An alternative approach described by [15] consists in first using transposition to matricize each tensor in the calculation, then applying a fast GEMM kernel, and transposing again for the final result. This approach takes advantage of existing kernels but the transposition operations are total overhead. [30] attempt to avoid the transposition cost by performing the transpositions as data is loaded into shared memory. This in conjunction with specialized kernels avoids the overhead of transposition. Separately [29] avoid transposition by developing `stridedBatchedGemm` for single-mode contractions. In effect looping over GEMM operations without reshaping the tensor; which can experience poor memory access patterns if the tensor modes are highly rectangular. Recently [13] exploit domain specific properties of data reuse in tensor contractions, applying their insights to devise an explicit code generator and demonstrate superior performance on most test cases of the TCCG benchmark.

None of these tensor contraction works support *generalized tensor operations*, and in particular do not generalize to operations between more than two tensors, nor support convolution along any tensor mode. Therefore, our work is unique and novel.

8 Conclusion

The fused CP convolution has considerable potential as a viable and high performing operation in future deep learning tasks, particularly for tensorial neural networks. The runtime performance is superior to the current baseline for exactly the types of low-rank approximations we expect to be of most interest to neural network researchers. The reduction in global memory usage is considerable when compared to both a traditional convolutional layer, and an optimal pairwise sequential evaluation. We speculate that the alleviation of global memory limits will enable researchers to find other novel ways to use the available memory, for example, on larger training batches.

The fusion approach is not without downsides however. Foremost is the required engineering and development time. Next, the lack of flexibility for use in other network architectures based on alternative tensor decompositions. The fused CP convolution is not applicable to either a Tucker or Tensor Train decomposition. All alternative *tensorial neural network* layers that employ *generalized tensor operations* would need custom fused operators.

These drawbacks are not shared with the optimal pairwise sequencer approach. By taking advantage of existing library functions and their gradient operations, current TNN training schemes can proceed with existing implementations,

while potentially benefiting from a reordering of pairwise evaluations. Unfortunately, even when optimally sequenced, pairwise evaluation must store intermediate values, limiting use of scarce GPU global memory.

Source code for this work including implementations and benchmark tests is available at https://github.com/Areustle/ParallelTNNLayers.

References

1. Anandkumar, A., Ge, R., Hsu, D., Kakade, S.M., Telgarsky, M.: Tensor decompositions for learning latent variable models. J. Mach. Learn. Res. **15**, 2773–2832 (2014)
2. Auer, A.A., Baumgartner, G., Bernholdt, D.E., Bibireata, A., Choppella, V., Cociorva, D., Gao, X., Harrison, R., Krishnamoorthy, S., Krishnan, S., Lam, C.-C., Lu, Q., Nooijen, M., Pitzer, R., Ramanujam, J., Sadayappan, P., Sibiryakov, A.: Automatic code generation for many-body electronic structure methods: the tensor contraction engine. Mol. Phys. **104**(2), 211–228 (2006)
3. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282 (2017)
4. Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., Shelhamer, E.: cuDNN: efficient primitives for deep learning. arXiv:1410.0759 [cs], October 2014. arXiv: 1410.0759
5. Cichocki, A., Lee, N., Oseledets, I.V., Phan, A.H., Zhao, Q., Mandic, D.: Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: perspectives and challenges part 1. arXiv preprint arXiv:1609.00893 (2016)
6. Cichocki, A., Lee, N., Oseledets, I.V., Phan, A.H., Zhao, Q., Mandic, D.P.: Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: perspectives and challenges PART 1. CoRR, abs/1609.00893 (2016)
7. Cichocki, A., Phan, A.-H., Zhao, Q., Lee, N., Oseledets, I., Sugiyama, M., Mandic, D.P., et al.: Tensor networks for dimensionality reduction and large-scale optimization: part 2 applications and future perspectives. Found. Trends® Mach. Learn. **9**(6), 431–673 (2017)
8. Nvidia Corporation. Nvidia Turing GPU Architecture (2018). https://nvidia.com/en-us/geforce/news/geforce-rtx-20-series-turing-architecture-whitepaper. Accessed 09 Sept 2019
9. Denton, E.L., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in Neural Information Processing Systems, pp. 1269–1277 (2014)
10. Abadi, M., et al.: Dean, Tucker, Yu, and TensorFlow: Large-scale machine learning on heterogeneous systems (2015). tensorflow.org
11. Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. GAMM-Mitteilungen **36**(1), 53–78 (2013)
12. Janzamin, M., Sedghi, H., Anandkumar, A.: Generalization bounds for neural networks through tensor factorization. CoRR, abs/1506.08473 (2015)
13. Kim, J., Sukumaran-Rajam, A., Thumma, V., Krishnamoorthy, S., Panyala, A., Pouchet, L.-N., Rountev, A., Sadayappan, P.: A code generator for high-performance tensor contractions on GPUs. In: 2019 IEEE/ACM International Symposium on Code Generation and Optimization (CGO), Washington, DC, USA, pp. 85–95. IEEE, February 2019

14. Knuth, D.E.: The Art of Computer Programming, Volume 1 (3rd edn.): Fundamental Algorithms. Addison Wesley Longman Publishing Co., Inc., Redwood City (1997)
15. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
16. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
17. Kossaifi, J., Khanna, A., Lipton, Z., Furlanello, T., Anandkumar, A.: Tensor contraction layers for parsimonious deep nets. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1940–1946. IEEE (2017)
18. Kossaifi, J., Lipton, Z.C., Khanna, A., Furlanello, T., Anandkumar, A.: Tensor regression networks. CoRR, abs/1707.08308 (2017)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
20. Lam, C.-C., Sadayappan, P., Wenger, R.: On optimizing a class of multi-dimensional loops with reductions for parallel execution. Parallel Process. Lett. **7**(2), 157–168 (1997)
21. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., Lempitsky, V.: Speeding-up convolutional neural networks using fine-tuned CP-decomposition. arXiv preprint arXiv:1412.6553 (2014)
22. Li, J., Sun, Y., Su, J., Suzuki, T., Huang, F.: Understanding Generalization in Deep Learning via Tensor Methods (2020)
23. Ma, W., Krishnamoorthy, S., Villa, O., Kowalski, K.: GPU-based implementations of the noniterative regularized-CCSD(T) corrections: applications to strongly correlated systems. J. Chem. Theory Comput. **7**(5), 1316–1327 (2011)
24. Novikov, A., Podoprikhin, D., Osokin, A., Vetrov, D.P.: Tensorizing neural networks. CoRR, abs/1509.06569 (2015)
25. Orús, R.: A practical introduction to tensor networks: matrix product states and projected entangled pair states. Ann. Phys. **349**, 117–158 (2014)
26. Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
27. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)
28. Pfeifer, R.N.C., Haegeman, J., Verstraete, F.: Faster identification of optimal contraction sequences for tensor networks. Phys. Rev. E **90**(3), 033315 (2014). arXiv:1304.6112
29. Shi, Y., Niranjan, U.N., Anandkumar, A., Cecka, C.: Tensor contractions with extended BLAS kernels on CPU and GPU. In: 2016 IEEE 23rd International Conference on High Performance Computing (HiPC), pp. 193–202 (2016)
30. Springer, P., Bientinesi, P.: Design of a high-performance GEMM-like Tensor-Tensor Multiplication. CoRR (2016)
31. Su, J., Li, J., Bhattacharjee, B., Huang, F.: Tensorial neural networks: generalization of neural networks and application to model compression. CoRR, abs/1805.10352 (2018)

Budget Active Learning for Deep Networks

Patrick Kinyua Gikunda[1,2(✉)] and Nicolas Jouandeau[2]

[1] Computer Science Department, Dedan Kimathi University of Technology, Nyeri, Kenya
kinyuagikunda@gmail.com
[2] Computer Science Department, Paris8 University, Saint-Denis, France
n@up8.edu

Abstract. In the digital world unlabeled data is relatively easy to acquire but expensive to label even with use of domain experts. On the other hand, state-of-the-art Deep Learning methods are dependent on large labeled datasets for training. Recent works on Deep Learning focus on use of Active Learning (AL) with uncertainty for model training. Although most uncertainty AL selection strategies are very effective, they fail to take informativeness of the unlabeled instances into account and are prone to querying outliers. In order to address these challenges, we propose a *Budget Active Learning* (BAL) algorithm for Deep Networks that advances active learning methods in three ways. First, we exploit both the uncertainty and diversity of instance using uncertainty and correlation evaluation metrics. Second, we use a budget annotator to label high confidence instances, and simultaneously update the selection strategy. Third, we incorporate AL in Deep Networks and perform classifications on untrained and pretrained models with two classical and a plant-seedling sets of data while minimizing the prediction loss. Experimental results on the three datasets of varying sizes demonstrate the efficacy of the proposed BAL method over other state-of-the-art Deep AL methods.

Keywords: Active learning · Budget learning · Deep network

1 Introduction

Current ICT technologies include Internet of Things [1], Remote Sensing [2], Cloud Computing [3] and Big Data [4]. The continuous use of these technologies to collect, monitor, measure, store and analyze data has led to a phenomena of Big Data [5] which is in abundance of unlabeled data. Unlabeled data is relatively easy to acquire and it is expensive to label even with use of domain experts. For example, its expensive to hire dermatologists to annotate 129,450 skin cancer images [6]. Even when using state-of-the-art computing resources, training a Machine Learning (ML) model on large data sets can take long time. However, like other ML researchers [7], we believe that *ML algorithm does not need all of*

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 488–504, 2021.
https://doi.org/10.1007/978-3-030-55180-3_36

the available data for training. The main motivation for use of *Active Learning* (AL) is that, if a learning algorithm can pick the data it want to learn from, then a small set of selected data-points can be used for training. Typically this process would involve randomly sampling large amount of data from underlying distribution for training a model. Collecting large amount of labeled data for training is time consuming and expensive. AL provides methods for analyzing vast amount of data with improved efficiency than other computing approaches, because of the ability to iteratively select the most informative data samples [8]. AL is a semi-supervised method meaning that it does not require labels of all the samples in a dataset. In unsupervised methods no labeled samples are used and for fully supervised all samples are labeled. The decision of how much data to use for training a Deep Learning model or alternatively the level of performance required is a resource management decision.

The emphasis in AL is to evaluate the informativeness of an instance, with an assumption that an instance with higher classification uncertainty is more crucial to label. This classical approach usually uses statistical theory such as entropy and margin to measure instance utility, however it fails to capture the data distribution information contained in the unlabeled data. This can eventually cause the classifier to select outlier instances to label Therefore, its important to consider the classification uncertainty as well as instances diversity in a population while developing an AL solution. In this paper, we present a *Budget Active Learning* (BAL) for Deep Networks a new robust AL method created by combining both uncertainty and correlation measure as an instance informativeness evaluation metric. An instance is selected based on its informativeness measure and then a budget annotator is used to label the instance. After each successful labeling the model selection strategy is updated with the new labeled data information. We perform various experiments on batched SVHN, CIFAR10 and plant-seedling-V2 datasets using Deep Networks models: Inception-V3, DenseNet and SqueezeNet.

The rest of the paper is organized as follows: Section 2 highlights related works. Section 3 presents our proposed Budget AL algorithm. Section 4 presents the experiments and results. Section 5 concludes the paper.

2 Literature Review

Successful investigations on ways to reduce labeling cost by use of AL has been going on for years now [9]. AL helps reduce the training data by selecting the most informative instances to label for training the model [10]. In a typical AL method, learning proceeds sequentially, while actively querying the labels of some instances from the membership queries. In AL there are three scenarios in which the ML algorithm will query the label of an instance, they include: a) *Membership Query Synthesis* that generates constructs of an instance from underlying distribution [11]; b) *Stream-Based Selective Sampling* that uses query strategy to determine whether to query the label of an instance or reject it based on informativeness [12]; c) *Pool-Based Sampling* that uses instances that are

drawn from a pool of unlabeled data according to some informativeness measure [13]. Many recent works focuses on use of pool-based sampling approach. The aim is to query labels of the most informative instances, consequently reducing labeling costs and accelerating the learning.

In recent time, there are a number of works focusing on AL strategies to reduce the labeling cost. Yang *et al.* (2017) defines in [14] ways to segment biomedical images by combining fully convolutional network and AL to reduce annotation effort by making suggestions on the most effective annotation areas. In their approach, the network is used to provide uncertainty and similarity information which is used to evaluate the most informative areas for annotation. Sener *et al.* (2017) in [15] defines AL as a core-set selection problem by choosing a set of points that the model can use to learn in a batch setting environment. A geometrical method is used to characterize the performance of the selected subset. Wanh *et al.* (2016) in [16] introduced a framework for updating the feature representation and the classifier simultaneously. A sample selection strategy is used to improve the classifier performance while reducing the manual annotation. Huang *et al.* (2018) in [17] uses fine tuned pretrained model on most useful examples. The examples are estimated based on potential contribution of an instance to feature representation. Iscen *et al.* (2019) in [18] introduces a transductive method that uses nearest neighbor graph to make predictions for generating pseudo-labels of the unlabeled dataset. Wang *et al.* (2014) in [19] combines AL and transfer learning into a Gaussian process based approach, and sequentially uses predictive covariance to select most suitable queries from the target domain. In their study Kale and Liu in [20] uses a combination of AL and Transfer Learning to learn labeled data from source domain for classification in target domain. Kale *et al.* (2015) in [21] introduces a framework for generating effective label queries by performing transfer learning. The framework is able to perform both the un-supervised and semi-supervised learning. Cai *et al.* (2019) in [22] defines online video recommendation as a multi-view AL problem and they proposed a framework to learn the mapping from visual view to text view. In their work Joshi *et al.* (2009) proposes an uncertainty measure that generalizes margin based uncertainty to the multi-class [23]. Chakraborty *et al.* (2011) propose a dynamic-batch-mode-AL combined with selection criteria as a single formulation [24].

The conventional way to reduce the cost of designing Deep Learning model and optimizing its parameters is by exploiting available pretrained models. Use of pretrained models trained on large benchmark dataset can helps reduce the training cost by utilizing the learned information. This is also referred to as *Transfer Learning* (TL) [25]. In TL, instead of starting the learning process from randomly initialized model weights, learning starts from patterns that have been learned when solving a different problem. This way there is leverage on previous learnings. The information transfer between the source and the target domain is done through feature sharing [26] and components transformation [27]. Performing batch training with gradient descent optimization helps address the challenge of limited computing power in deep learning. However, it is not possible

to train Deep Networks efficiently with large training set. To overcome this challenge, a mini-batch gradient descent is performed by splitting the training set into smaller sets and gradient descent is implemented on each of the batches. This approach make training more faster and efficient. Classical state-of-the-art Deep Network models include: AlexNet [28], NIN [29], ENet [30], ZFNet [31], GoogLeNet [32] and VGG 16 [33]. Modern models include: Inception [34], ResNet [35], and DenseNet [36]. These networks have achieved impressive performance on computer vision, speech and text recognition with effective representations for visual objects [37].

From the literature presented, recent AL works focus on selecting a single informative unlabeled instance to label using uncertainty metrics. One main shortcoming of the above approaches is poor generalization for unseen instances in the domain. This is due to the fact that they only select queries based on how the instance related to the classifier while ignoring unlabeled instances. Also with a large set of instances classification response time can be slow, therefore use of budget annotator will help reduce active selection and labeling time.

3 AL with Budget Annotation

In this section, we first describe the general AL algorithm, then we introduce our algorithm detailing each component. The following notation will be used in this paper. Let x represents instances and y represents labels, $D = D^L \cup D^U$, D^L denotes labeled instances where $D^L = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, D^U denotes unlabeled instances where $D^U = \{(x_1, ?), (x_2, ?), ..., (x_n, ?)\}$, D^H denotes high confidence instances and Θ denotes the model defined by model parameters. For label space L^S with m classes in D the label of D^U can be expressed as $y_i = l, l \in \{1, 2, ..., m\}$. Therefore, instance selection criteria in this study will be based on probability of x_i belonging to l^{th} class which can be expressed as:

$$p(y_i = l | x_i; \theta) \tag{1}$$

where θ denotes the CNN network weights and Eq. 1 denotes the network softmax output for l^{th} class.

Fu *et al.* presents a survey on instance selection and introduces in [38] an inefficient general AL algorithm for *Deep Networks*. We present in Algorithm 1 a new generalized form of AL. From lines 4 to 10, the model is iteratively defined according to a budget m.

Figure 1 describes the conceptual representation of our method. The method progressively get data as input from the unlabeled set. While on initial model parameters, the most informative instance is selected from the unlabeled set for labeling by the Budget Annotator. On successful selection and labeling the labeled instance is added to the training set and the model selection strategy is simultaneously updated and validated. Most informative samples and the classified samples are applied on the classifier output. The process to select and label instances will iterate until the budget is achieved while simultaneously updating the selection strategy.

Algorithm 1: General Active Learning.

Input: labeled instance set D^L, unlabeled instance set D^U, a budget m;
Output: a model Θ;
$\Theta \leftarrow$ `getModel`(D^L);
while $|D^L| < m$ **do**
  $D^U \leftarrow D \setminus D^L$;
  for *each* x_i *in* D^U **do**
    $u_i \leftarrow u(x_i, \Theta)$;
  $x^* \leftarrow \underset{i}{\text{argmax}}(u_i)$;
  $D^L \leftarrow D^L \cup \{x^*\}$;
  $\Theta \leftarrow$ `getModel`(D^L);
return Θ;

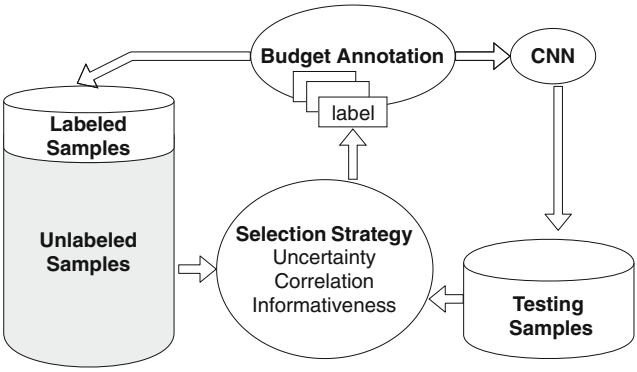


Fig. 1. BAL conceptual representation

In order to avoid the problem of generalization of unseen instances and to learn an accurate model, we present a robust approach by combining the strengths of different learning strategies. An AL annotator use evaluation metrics to compute the instance utility in order to select the most appropriate instance to label. The utility metrics considered in this work are uncertainty, correlation and informativeness measure, thus we present four main components: a) an uncertainty measure, b) correlation measure, c) an informative measure and d) and budget annotator.

3.1 Uncertainty Measure

Given a label space L^S the uncertainty measure f_u of a sample (features & label) can be defined as:

$$f_u(x) : \begin{cases} L^S \rightarrow R, & \text{(i) features view} \\ (D^U \times L^S) \rightarrow R, & \text{(ii) features-label view} \end{cases} \tag{2}$$

to a real number space R. From Eq. 2, (i) the uncertainty measure is computed from sample features only while (ii) the measure is computed from both the sample features and label. In our method we consider the uncertainty measure computed from sample features and label view which is considered the most effective [39]. Out of the three common uncertainty measure criteria namely least confidence, sample margin and entropy, we considered sampling margin since it integrates the second most probable class label in the uncertainty metric hence able to reduce the error rate by defining the decision boundary. We therefore define uncertainty measure as:

$$f_u(x_i) = p(y_i = l_1|x_i;\theta) - p(y_i = l_2|x_i;\theta) \tag{3}$$

High uncertainty value f_u implies current model have little knowledge of the instance, and including it into the training set can help improve the prediction performance of the model.

3.2 Correlation Measure

When developing efficient AL methods, its is critical to consider samples distribution information [40]. The instance diversity information aids in selecting most representative instances. In order to have more information about the unlabeled instances its appropriate to select a candidate instance in a more dense region. In addition, selecting an instance to label only based on uncertainty measure may lead to redundancy, therefore exploiting sample instance diversity will provide an optimal instance to label. Our method is based on the fact that the trade off between instance uncertainty and correlation is an essential AL problem to address. Given a label space L^S, we can define different groups of correlation of an instance x in a set of unlabeled set as;

$$f_c(x) : \begin{cases} D^U \times D^U \rightarrow R, & \text{feature view} \\ L^S \times L^S \rightarrow R, & \text{label view} \\ (D^U, y) \times (D^U, y) \rightarrow R, & \text{combined view} \end{cases} \tag{4}$$

to a real number space R. In Eq. 4, the combination of feature and label correlation is called combined view. Different algorithms exist for exploiting this type of combination [39]. Majorly these algorithms are used in a multi-label learning tasks when an instance has more than one label. This setting is ideal for mining tasks on instances with complex structure. On our work we focus on exploiting the pairwise similarities of instances, therefore the informativeness of an instances is weighed by average similarity to its neighbours. Let x_i and x_j be a pair of instances. To cope with the drawback of uncertainty based selection, we then consider the diversity by evaluating the correlation of the instances. Given a label space L^S the correlation measure $f_c(x_i, x_j)$ between a pair of instances in a sample x_i and x_j can be defined as:

$$f_c(x_i) = \frac{1}{D^U} \sum_{x_j \in D^U} (x_i, x_j) \tag{5}$$

The value of $f_c(x_i)$ represents the instance density of x_i in the unlabeled set. The larger the value, the more densely an instance is correlated with others. A low value of the correlation measure indicates an outlier instance which should not be considered for labeling.

3.3 Informativeness Measure

Our motivation is that the most representative instances of a distribution can be very informative for improving the generalization performance. Therefore, given correlation measure $f_c(x_i)$ and uncertainty measure $f_u(x_i)$ the informativeness of an instance can be defined as:

$$f_i(x) = f_u(x_i) \times f_c(x_i) \tag{6}$$

It can be rewritten as:

$$x^* = \underset{i}{\operatorname{argmax}}(u_i.c_i) \tag{7}$$

3.4 Instance Evaluation and Budget Labeling

Instance evaluation is based on the instance informativeness in a set. In our method we use query by a single model evaluation learned from the training set. The model is trained on all labeled instances: feature and label views. After querying for an unlabeled instance, a model prediction result is generated based on output probability distribution. Each instance $x_i = \{f_1^i, f_2^i, ...f_q^i, y^i\}$ in labeled set $D^L = \{x_1, x_2, ...x_s\}$ is represented in a feature space F consisting of a feature space and its class label y^i. The size of D^L is denoted by s and x_i denoted the ith instance in D^L. The prediction can be denoted as a mapping function from the feature space F to the class label space Y which can be expressed as;

$$p(x) : F \mapsto Y \tag{8}$$

The query strategy used in this work is based on the value of f_i discussed in Eq. (6). Instances are ranked based on the value f_i with top ranked instances being the most appropriate to label. Budget annotator is used to pick classes which has maximum predicted probability as if they were true labels. For CNN implementation we use entropy regularization, this way we are able to separate low density between classes. High confidence samples from D^H are selected and then assign predicted labels to them. For l^{th} category we define the budget label y_i as follows;

$$y^* = \underset{i}{\operatorname{argmax}}(p(y_i = l|x_i; \theta_{x,y})) \tag{9}$$

Under the current distribution $p(y_i = l|x_i; \theta)$ each possible instance $(x_1, ?)$ from the selected instances D^H will be labeled with label y_i. When $y_i = 1$, x_i is regarded as a high confidence sample. The model update strategy is to learn a model based on the information provided by model weights computed from

Algorithm 2: Efficient Budget Active Learning (BAL).

1 **Input:** labeled instance set D^L, unlabeled instance set D^U, a budget m;
2 **Output:** model Θ;
3 $\Theta \leftarrow$ `getModel` (D^L);
4 **while** $|D^L| < m$ **do**
5 **for** *each* x_i *in* D^U **do**
6 $u_i \leftarrow f_u(x_i)$;
7 $c_i \leftarrow f_c(x_i)$;
8 $x^* \leftarrow \underset{i}{\text{argmax}}(u, c)$;
9 $D^H \leftarrow \emptyset$;
10 **for** *each* i *in* D^U **do**
11 $x \leftarrow \underset{i}{\text{argmax}} f_i(x)$;
12 $D^H \leftarrow D^H \cup \{x\}$;
13 $y_i \leftarrow$ `getLabel`(D^H);
14 $D^U \leftarrow D^U \setminus \{y_i\}$;
15 $D^L \leftarrow D^L \cup \{y_i\}$;
16 $\Theta \leftarrow$ `getModel` (D^L);
17 **return** Θ;

model validation of the performance. The Algorithm 2 describes the *Budget Active Learning* (BAL) with budget annotation.

BAL is designed to train a classification model using a small labeled population sample proportion. At first the BAL is trained using the initial set of labeled data D^L, using the initial weights for pretrained models and random initialized weights for untrained models. In Algorithm 2, the labeling is defined by the budget m with model updates after each iteration (lines 4–16). Instance evaluation is done to identify the most informative and representative instance to label (lines 5–8). This evaluation returns the high confidence instances D^H selected from the unlabeled population (lines 10–12). For each of the selected instance, its label is queried and consequently the labeled set is updated. The model selection strategy is updated with the learned parameters after every iteration.

4 Experiments

To examine the efficiency of the proposed algorithm, we have considered public available datasets and state-of-the-art models.

4.1 Datasets

Three public available datasets namely CIFAR10 [41], Street View House Numbers (SVHN) [42] and plant-seedling-V2 [43] datasets are used. The statistical information of the datasets are summarized in Table 1. For large datasets

(CIFAR10 and SVHN), in regards to their size, we split the data into two sets; 20% as labeled set and 80% as unlabeled set. Half of the labeled set is randomly sampled as the training set, and the remaining as the validation set. The testing samples for each of the dataset is as shown in the table. For the other dataset (plant-seedling-V2), due to its very small size, 40% was randomly sampled as labeled set and 70% as the unlabeled set. In both cases, we tried to minimize the size of the training data, in order to demonstrate the efficiency of our budget AL method. For all data input, resize and normalize transformation was done in order to match the models input sizes and shapes.

4.2 Fine-Tuning Network Parameters

In order to suite the pretrained network to the dataset classes, the last layer (softmax layer) is truncated and replaced with a layer that matches the dataset classes. Back propagation is performed to fine-tune the pretrained weights. 10 model updates were carried out with a training batch size of 32 and a learning rate of 0.05. The number of model updates is sufficient to demonstrate the classification performance and efficiency of our method over the other methods. The training rate is carefully considered to ensure a good training stability and generalization is achieved.

Table 1. Selected datasets used in this work

Data	# Instance	# Label	# Train	# Validation	# Testing
CIFAR10 [41]	50k	10	5k	5k	10k
SVHN [42]	73k	10	7k	7k	26k
Plant-seedling-V2 [43]	6539	12	1k	807	807

4.3 Models

Table 2 presents six state-of-the-art Deep Networks models that have comparative few model parameters (M) expressed in million. While Deep Networks provide state-of-the-art prediction accuracy to many Machine Learning tasks, it comes at a high computational cost [44]. Model with more parameters (i.e. bigger networks and learnable parameters) is slower than a model with less parameters. The experiments were done using three of these models which have achieved best performance in ILSVRC and have lower parameters number (Inception-V3, DenseNet-169 and SqueezeNet). Instead of only training an entire CNN from scratch (with random initialization) we considered also transfer learning in order to leverage the training and then use ConvNet as an initialization and a fixed feature extractor for the task. In our experiments, we have used pretrained models given by Pytorch v1.3.0. In this section we will briefly discuss the architectures of the selected models.

Table 2. Deep networks models comparison on ImageNet [47].

Model	Input size	M	Top-1 acc (%)	Top-5 acc (%)
Inception-V1 [32]	224×224	5	70	90
Inception-V2 [45]	224×224	5	74	92
Inception-V3 [34]	$\mathbf{299 \times 299}$	**24**	**78**	**94**
Inception-V4 [48]	299×299	46	80	95
DenseNet-169 [36]	$\mathbf{224 \times 224}$	**14**	**76**	**93**
SqueezeNet [49]	$\mathbf{224 \times 224}$	**3**	**68**	**88**

GoogLeNet, a 2014 ILSVRC winner, was inspired by LeNet but implemented a novel Inception module. Their Inception module performs series of convolutions at different scales and subsequently concatenate the results. The module is built with several small convolutions. There has been tremendous efforts done to improve the performance of this architecture: a) Inception-V1 [45] has 3 different sizes of filters ($1 \times 1, 3 \times 3, 5 \times 5$) and max pooling. The outputs are concatenated and sent to the next Inception module; b) Inception-V2 [45] and Inception-V3 [34] factorize 5×5 convolution to two 3×3 convolution operations to improve computational speed. A 5×5 convolution is 2.78 times costly than a 3×3 convolution; stacking two 3×3 convolutions leads to a boost in performance; c) In Inception-V4 and Inception-ResNet the initial set of operations were modified before introducing the Inception blocks. The Fig. 2, 3 and 4 show the prediction accuracy comparison between our approach and other baseline methods on previously cited models.

When Deep Networks start converging, then degradation become apparent challange to performance. Thats means as the network depth increases, the accuracy gets saturated and then degrades rapidly. Deep Residual Neural Network (ResNet), a logical extension of DenseNet [36] created by Kaiming *et al.* [35] introduced a novel architecture with insert shortcut connections. The connections turns the network into a residual network. This was a breakthrough which enabled the development of much deeper networks. The residual enables the network learn to adjust the input feature map. Following this intuition the authors proposed a pre-activation variant using the insert shortcut connections by the gradients flowing through the shortct connections to the earlier layes unimpended. Each ResNet block is either 2 layer deep or 3 layer deep. It achieves a top-5 error rate of 3.57% which outperforms human-level performance. DenseNet which is a logical extension of ResNet, brings improved efficiency by concatenating each layer feature map to every successive layer within a dense block [36]. This enables feature reuse within the network by allowing later layers within the network to directly leverage the features from earlier layers. Now the feature-maps of all preceding layers can be used as inputs, and its own feature-maps can be used as inputs into all subsequent layers, this helps alleviate the vanishing-gradient problem, feature reuse and consequently reduce number of parameters.

In recent times with use of Internet of Things and Cloud Computing, there is constant communication between the servers and the clients. This brings a need for a smaller sized model with similar or improved efficiency as the state of the art models. SqueezeNet [46] achieves AlexNet-level accuracy with 50x fewer parameters [47]. Additionally with model compression technique one can achieve 510 times smaller than AlexNet compression. In order to reduce the number of parameters by 9 times, a 3×3 filters are replaced with 1×1 filters. Subsequently, number of input channels is reduced to 3×3 filters. Finally, the feature map is down-sampled in order to have larger activation maps.

4.4 Results

The proposed approach was implemented on NVIDIA Tesla P100 GPU. Using few model update iterations, our method demonstrates impressive prediction accuracy over the other Deep AL methods.

In the experiments losses and accuracies per model update were monitored while comparing the following Deep Active Learning baseline methods:

- **Budget AL (BAL)**: our proposed method.
- **Core-Set AL (CSAL)**: method proposed in [15] which defines the AL problem as a competitive sample core-set selection which is then applied to a CNN in a batch setting.
- **Deep Bayesian Active Learning (DBAL)**: a Bayesian framework proposed in [8] for high dimensional data which considers Deep Learning problem of dependence on big amount of data.

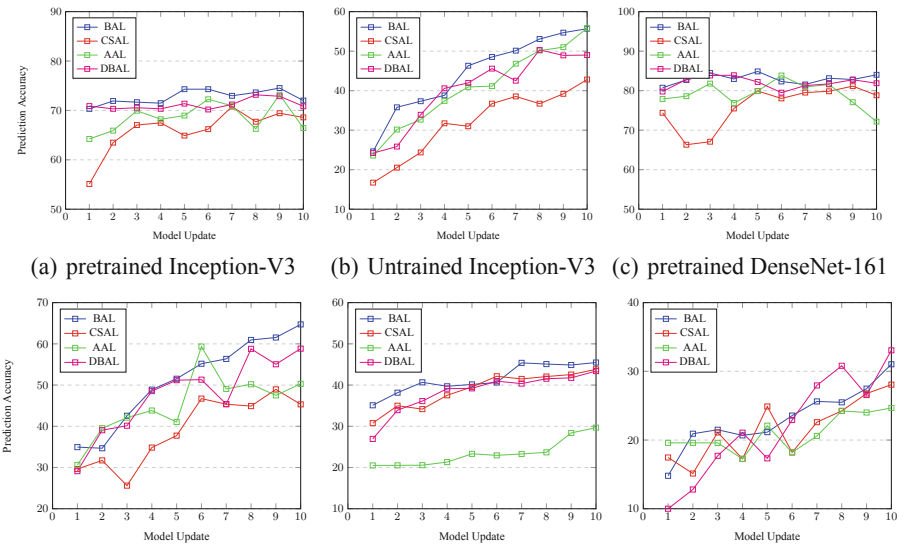


Fig. 2. Prediction comparison on CIFAR10 dataset.

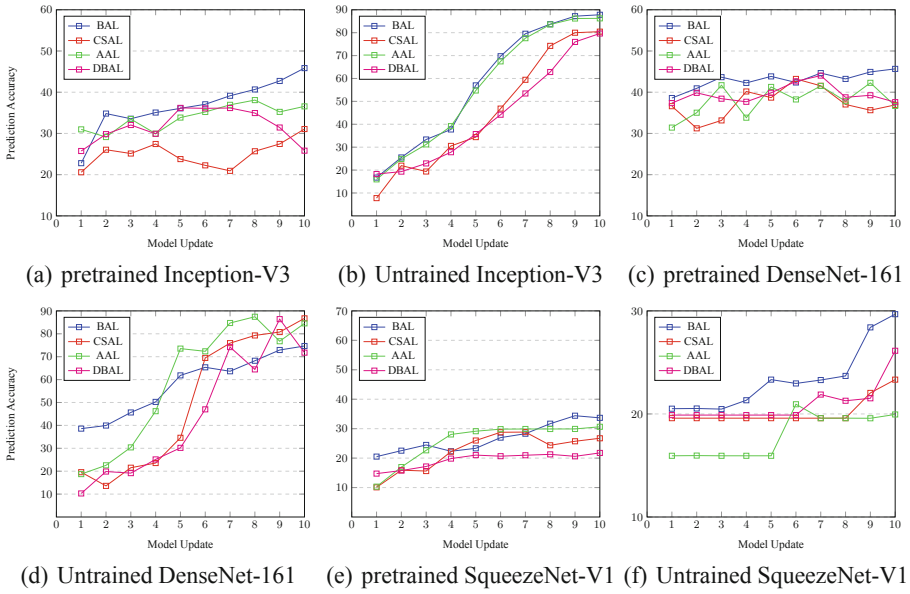


Fig. 3. Prediction comparison on SVHN dataset.

– **Adversarial AL for Deep Networks (AAL)**: a margin based approach proposed in [50] for Deep Networks with intention of reducing the number of queries to the oracle during training.

Impressive performance is recorded by the methods on the pretrained models as compared to the un-trained models. In general, from the results the pretrained DenseNet and Inception models on CIFAR10 leverage much better than SqueezeNet on same dataset. This means that the model weights for DenseNet and Inception model leverage better that those of SqueezeNet to this type of dataset. On all the training instances, BAL performs better than all other baseline active learning methods as shown in Fig. 2. On the un-trained models the prediction performance seem to edge up as the model selection strategy gets updated.

On SVHN dataset, all Deep Active Learning methods performs poorly except on un-trained DenseNet and Inception models. The performance on these models improves after the fifth model update. This is so because initially the models weights do not perform well with this dataset but after several self tuning there is improved prediction accuracy. Following the poor performance exhibited in SqueezeNet on both CIFAR10 and SVHN data, we did not conduct experiments with SqueezeNet on the plant-weed detection problem.

Plant Weed Detection. Agriculture is a critical for human survival and it remains a major driver of many economies around the world. With increase

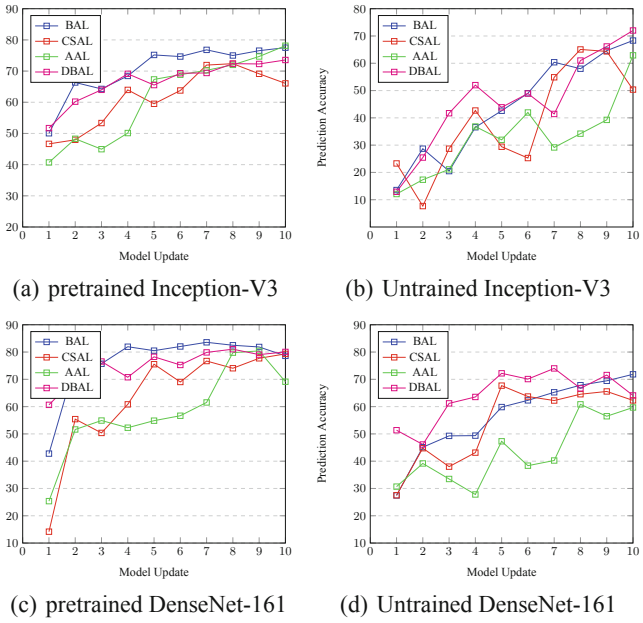


Fig. 4. Prediction comparison on plant-seedling-V2 dataset.

demand for food and other agricultural production challenges, there is sure need to improve on production output. Current agricultural machine vision solutions are faced with accurate and reliable large scale weed detection. In this section we present a plant seedling weed detection problem using a plant-seedling-V2 dataset [43]. The plant-seedling dataset contains 6539 images from 960 RGB unique seedlings of plants belonging to 12 species at different growth stages with a physical resolution of 10 pixels/mm. Because of small dataset available, 15% of the set was used for training our algorithm for weed identification and 12% used for validation, the rest of the data used as unlabeled dataset. In addition, in an effort to avoid overfitting the convolutional base of the networks was frozen and its output used in the classifier.

In Fig. 4(a) and 4(c) we compare our method with other Deep Active Learning methods in both pretrained Inception-V3 and DenseNet-161 models on plant-seedling dataset. The results indicate the efficiency of our method as compared to performance of other Deep AL methods discussed. Our method is able to adapt better with the pretrained parameters and quickly provide better prediction. Figure 4(b) and 4(d) show the performance on the untrained versions of the models on the same dataset. Using the initial pretrained parameters to initialize the models yields to better prediction accuracy quickly within few model updates. The main results are shown in Fig. 2 and 3. Overall, BAL (in blue line) is able to outperform other Deep AL methods on major datasets including the plant-seedling dataset. By comparing our method with other methods, we

notice an apparent increase in classification accuracy which indicate that using both instance uncertainty and correlation measure is more efficient. BAL is able to pick the most representative candidate point from the unlabeled population. In addition, from the plant-seedling shown in Fig. 4, we observe the superiority of our method tends to be more obvious even when the number of instances is small. Its clear that our method can generalize better than other discussed methods by selecting the most representative instances with only few queries.

5 Conclusion

The main objective of AL is to label the most informative instance in order to achieve high prediction accuracy with minimum cost. Use of AL in recent technologies is an active research area with efforts to improve on the prediction accuracy while using less data. In this paper, we propose a BAL method for cost-effective training of Deep Networks. Instead of training from scratch with random initialization, a pretrained model parameters can be used to initialize a model to a new target task by fine tuning with a few actively queried examples, thus significantly reducing the cost of designing the network architecture and cost of labeling a large training set. Using BAL, classification task was able to record 85% prediction accuracy quickly using fairly small amount (10 to 20% of data) of data as training data as compared to conventional AL methods on Deep Network Models. In the future, we plan to apply the approach on more real life datasets and more pretrained models. In addition, the feature transformation on each layer will be further studied while considering different types of data input.

References

1. Weber, R.H., Weber, R.: Internet of Things, vol. 12. Springer, Heidelberg (2010)
2. Ray, A.S.: Remote sensing in agriculture. Int. J. Environ. Agric. Biotechnol. **1**(3), 238540 (2016)
3. Jinbo, C., Xiangliang, C., Han-Chi, F., Lam, A.: Agricultural product monitoring system supported by cloud computing. Cluster Comput. **22**(4), 8929–8938 (2019)
4. Chi, M., Plaza, A., Benediktsson, J.A., Sun, Z., Shen, J., Zhu, Y.: Big data for remote sensing: challenges and opportunities. Proc. IEEE **104**(11), 2207–2219 (2016)
5. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
6. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature **542**(7639), 115 (2017)
7. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 2208–2217 (2017)
8. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1183–1192 (2017)

9. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Mach. Learn. **15**(2), 201–221 (1994)
10. Settles, B.: Active learning literature survey. University of Wisconsin-Madison, Department of Computer Sciences (2009)
11. Angluin, D.: Queries and concept learning. Mach. Learn. **2**(4), 319–342 (1988)
12. Zhu, X., Zhang, P., Lin, X., Shi, Y.: Active learning from data streams. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 757–762 (2007)
13. Nigam, K., McCallum, A.: Pool-based active learning for text classification. In: Conference on Automated Learning and Discovery (CONALD) (1998)
14. Yang, L., Zhang, Y., Chen, J., Zhang, S., Chen, D.Z.: Suggestive annotation: a deep active learning framework for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 399–407. Springer, Cham (2017)
15. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. arXiv preprint arXiv:1708.00489 (2017)
16. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Trans. Circuits Syst. Video Technol. **27**(12), 2591–2600 (2016)
17. Huang, S.J., Zhao, J.W., Liu, Z.Y.: Cost-effective training of deep CNNs with active model adaptation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1580–1588 (2018)
18. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5070–5079 (2019)
19. Wang, X., Huang, T.K., Schneider, J.: Active transfer learning under model shift. In: International Conference on Machine Learning, pp. 1305–1313 (2014)
20. Kale, D., Liu, Y.: Accelerating active learning with transfer learning. In: 2013 IEEE 13th International Conference on Data Mining, pp. 1085–1090, December 2013
21. Kale, D., Ghazvininejad, M., Ramakrishna, A., He, J., Liu, Y.: Hierarchical active transfer learning. In: Proceedings of the 2015 SIAM International Conference on Data Mining, pp. 514–522. Society for Industrial and Applied Mathematics (2015)
22. Cai, J.J., Tang, J., Chen, Q.G., Hu, Y., Wang, X., Huang, S.J.: Multi-view active learning for video recommendation. In: Proceedings of IJCAI 2019, Macao, China (2019). https://www.ijcai.org/proceedings/2019/0284.pdf
23. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2372–2379 (2009)
24. Chakraborty, S., Balasubramanian, V., Panchanathan, S.: Dynamic batch mode active learning. In: CVPR 2011, pp. 2649–2656 (2011)
25. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2009)
26. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
27. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2010)
28. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
29. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)

30. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: a deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
31. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer, Cham (2014)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
35. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
36. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
37. Gikunda, P.K., Jouandeau, N.: State-of-the-art convolutional neural networks for smart farms: a review. In: Intelligent Computing-Proceedings of the Computing Conference, pp. 763–775. Springer, Cham (2019)
38. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowl. Inf. Syst. **35**(2), 249–283 (2013)
39. Huang, S.J., Gao, N., Chen, S.: Multi-instance multi-label a information regularization with partially labeled datactive learning. In: IJCAI, pp. 1886–1892 (2017)
40. Szummer, M., Jaakkola, T.S.: Advances in Neural Information Processing Systems, pp. 1049–1056 (2003)
41. Krizhevsky, A., Hinton, G.: Convolutional deep belief networks on CIFAR-10. **40**(7), 1–9 (2010, unpublished manuscript)
42. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
43. Giselsson, T.M., Jørgensen, R.N., Jensen, P.K., Dyrmann, M., Midtiby, H.S.: A public image database for benchmark of plant seedling classification algorithms. arXiv preprint arXiv:1711.05458 (2017)
44. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S.: Efficient processing of deep neural networks: a tutorial and survey. Proc. IEEE **105**(12), 2295–2329 (2017)
45. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
46. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360 (2016)
47. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
48. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial (2017). Intelligence

49. Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zao, S., Keutzer, K.: Squeezenext: hardware-aware neural network design. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1638–1647 (2018)
50. Ducoffe, M., Precioso, F.: Adversarial active learning for deep networks: a margin based approach. arXiv preprint arXiv:1802.09841 (2018)

Surface Defect Detection Using YOLO Network

Muhieddine Hatab(✉), Hossein Malekmohamadi, and Abbes Amira

Institute of Artificial Intelligence, De Montfort University, Leicester LE1 9BH, UK
muhieddine.hatab91@gmail.com,
{hossein.malekmohamadi,abbes.amira}@dmu.ac.uk

Abstract. Detecting defects on surfaces such as steel, can be a challenging task because defects have complex and unique features. These defects occur in many production lines and vary from one production line to another. In order to detect these defects, the You Only Look Once (YOLO) detector which uses a Convolutional Neural Network (CNN), is used and received only minor modifications. YOLO is trained and tested on a dataset containing six kinds of defects to achieve accurate detection and classification. The network can also obtain the coordinates of the detected bounding boxes, giving the size and location of the detected defects. Since manual defect detection is expensive, labor-intensive and inefficient, this paper contributes to the sophistication and improvement of manufacturing processes. This system can be installed on chipsets and deployed to a factory line to greatly improve quality control and be part of smart internet of things (IoT) based factories in the future. YOLO achieves a respectable 70.66% mean average precision (mAP) despite the small dataset and minor modifications to the network.

Keywords: YOLO · Defect detection · CNN · Computer vision · Transfer learning

1 Introduction

In every factory, defects can occur on products rolling out at the end of the conveyor line. This is due to many factors such as contamination, human error, machinery malfunctions and more. These defects include scratches and patches and not only is the defect purely cosmetic, in some cases it is structural and can cause damage to the steel surface such as corrosion, low wear resistance and short fatigue life which can lead to disastrous results where the products are meant to be used [1]. In order to show the importance of catching steel surface defects, tests are conducted on structural steel with and without defects and the results showed that metal surfaces with defects have 40% less strength with much faster strength degradation [2]. Safety is another crucial factor to consider since metal surfaces are used in all kinds of applications ranging from automotive applications all the way to construction.

To keep up with production line requirements, the designed defect detector must be accurate and fast. Factories these days have come a long way and work at a very high pace rolling hundreds of products out every hour. The detector must also be able to distinguish between defects and non-defective interference such as dust. Inspection and

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 505–515, 2021.
https://doi.org/10.1007/978-3-030-55180-3_37

quality assessment used to be done manually by humans who are prone to suffer from exhaustion and can be slower than machines. Moreover, training operators requires time, money and a difficult process finding people fit for the job and as mentioned earlier the usage of steel plates ranges over a large number of applications with some being critical and dangerous in the case of defects not being caught. Computer vision is helping in visual inspection and replacing manual labor in many industries [3].

CNNs are one of the best options for computer vision tasks and have allowed many advances in applications like image segmentation [4, 5] and the classification of objects [6, 7]. CNNs have also been used in industrial applications [8–10]. Moreover, CNNs have convolution layers that take care of feature extraction, they are rugged when it comes to shifts and distortions in the image, they require less memory, are easier to train and are better and faster due to the reduced number of parameters.

In this paper, the YOLO network is used for detection and classification of various defects in steel surfaces. The network is also able to extract the coordinates of the defects which in return gives the locations and sizes.

This paper is structured as follows: In Sect. 2, the background is presented, in Sect. 3, the methodology is explained including the training and testing process, Sect. 4 contains a discussion and analysis of the results and finally, Sect. 5 concludes this paper and mentions future work.

2 Background

2.1 YOLOV3 and Darknet-53

YOLO [11] is a one-shot object detection algorithm and is one of the fastest algorithms that exist today. It is mostly used in areas where speed is a crucial element, without the loss of too much accuracy. It uses a convolutional neural network which is essential when it comes to feature extraction. The way YOLO works is it divides an image into an S $\times$ S grid of cells where each cell is then responsible for predicting the presence of an object in it, P(object), as well as producing a number of bounding boxes which are likely to encompass objects. Each of the predicted bounding boxes has a confidence score where confidence is:

$$\text{P(Object)} \times \text{IOU(pred, truth)}$$

Intersection Over Union (IOU), an evaluation metric, is used to measure the difference between the ground truth bounding box which is already defined in the dataset and the bounding box predicted by YOLO (Fig. 1). The predicted bounding boxes that are closest to the ground truth are kept and their confidence scores are increased whereas the boxes that have a low IOU intersection with the ground truth are given a low confidence score. Five values are predicted by the network for each bounding box. (x, y) are the center of the bounding box and (w, h) are the width and height.

The next prediction is the conditional probability, P(Class|Object), where the probability of a certain class being in one of the bounding boxes is calculated. The final predictions are several bounding boxes scattered all around the image. YOLO thresholds these predictions using non-maximum suppression (NMS) to remove unwanted and

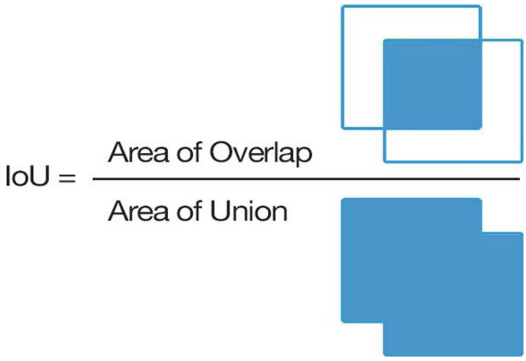


Fig. 1. Intersection over union [12]

duplicate bounding boxes. The network then ends up with only the necessary predictions shown on the image. Figure 2 shows an example of how the image is divided and each object is defined and at the end, surrounded by a bounding box.

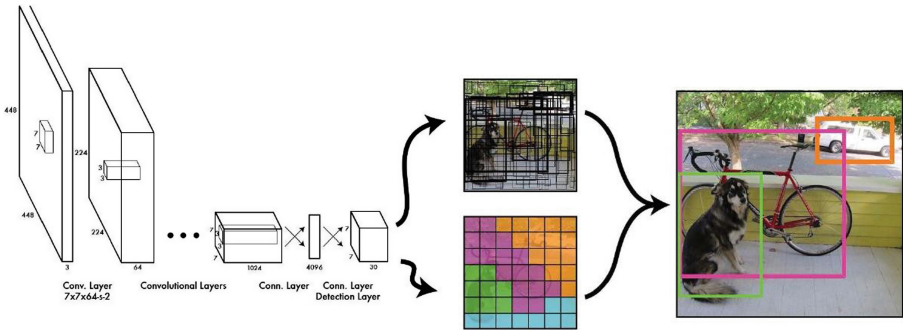


Fig. 2. YOLO network pipeline [13]

The backbone of YOLO is called DARKNET [14] created by Joseph Redmon, which is a neural network framework written in NVIDIA's Compute Unified Device Architecture (CUDA) and C. Its advantages are that it is quick, slim and easy to work with. Unlike its predecessor, YOLOv3 uses DARKNET-53 instead of DARKNET-19 where the former has 53 convolutional layers trained on ImageNet and is much deeper than the previous versions. It composes mainly of 3×3 and 1×1 filters with shortcut connections. It is also faster due to better utilization of the GPU. Darknet-53 is also proven to have better performance than ResNet-101 and it is 1.5 times faster and compared to ResNet152 it has similar performance but is 2 times faster [11]. DARKNET has its own commands and parameters which are used to train, test, calculate and perform many other operations on the model being worked on. This paper uses a slightly modified DARKNET53 by AlexeyAB [15] to allow for training on custom datasets.

2.2 Related Work

There are many methods for surface defect detection. In a paper [16], a simple CNN model is presented to detect defects in metal steel surfaces where the model achieved moderate results. However, with changes in the number of batches, as well as some data augmentation, 99% accuracy is achieved in training and testing. In [17], a two-layer convolutional network is proposed to detect surface defects where the loss function is calculated using categorical cross-entropy. After testing the system on testing images, the system is found to be 64.7% accurate which is acceptable given the small dataset. The disadvantage here is that there are only two convolutional layers which is not enough to extract features from a very small dataset. This issue is tackled in another paper [18] where it is decided to modify the YOLO detector to be fully convolutional where the network has 25 convolutional layers for feature extraction and 2 convolutional layers to predict the defect class and bounding box. With this architecture, the YOLO network is able to learn its own spatial downsampling instead of deterministic spatial downsampling. In this case, YOLO achieved a mAP of 97.55% and a recall rate of 95.86%. Another paper [19] which explored an approach for surface defect detection using deep learning used a twostage method which comprised of a segmentation network and decision network. The model worked fine and better than other approaches when experimenting on the Kolektor Surface-Defect Dataset (KolektorSDD); however, it still experienced 5 misclassifications and suffered a bit when it came to images with lower resolution. It did, however, achieve an accuracy of 99%. Table 1 shows a comparison of different approaches used to achieve the same goal as this paper with some of them using similar images and different performance metrics.

Table 1. Performance comparison of different methods

Model	Performance measure
CNN, Gathered dataset [16]	Acc: 99% mAP: N/A
2 Layer CNN, NEU surface defect database [17]	Acc: 64.7% mAP: N/A
Fully Convolutional YOLO, Gathered dataset [18]	Acc: N/A mAP: 97.55%
Segmentation + Decision Network, KolektorSDD [19]	Acc: 99% mAP: N/A

3 Methodology

Originally, YOLO is a pretrained object detector, trained to detect everyday objects such as tables, chairs, cars, phones and others. A modified version of YOLOV3 is used in this paper. Changes to the hyperparameters are made to be able to train and test using the custom dataset provided. The original dataset labels needed some modifications since YOLO only accepts a specific format and 5 specific parameters to associate the labels to the images and train properly.

3.1 Dataset

The images are obtained from the Northeastern University (NEU) surface database [20–22] which contains six types of defects (rolled-in scale (Rs), patches (Pa), crazing (Cr), pitted surface (Ps), inclusion (In) and scratches (Sc)) with 300 images for each defect (1800 total). Image size is 200 × 200 pixels with a.bmp format and the images are in grey-scale. The defects in the images vary and are provided in many shapes, sizes, illumination and orientation. The images are already labelled, and the labels contained information such as the location and size of the bounding box in an XML format. For this paper, the images are resized to 608 × 608 pixels using an online resizing tool [23] since the original size is too small. YOLO automatically resizes the input images to smaller dimensions when training, so it is crucial to start off with a somewhat large image so that the defects to be detected in the images are not too small, but closer in size to defects in images and videos provided by cameras in factories [24]. After many trials, it is found that 608 × 608 pixels is the best size and gave the best results. The labels are modified as well to fit the YOLO format since YOLO takes five values to produce the bounding boxes. Therefore, the results are 1800 text files each containing the five values in the following format: "(object-id) (xcentre) (y-centre) (width) (height)". The images are split into 10% for testing and 90% for training. It is important to note that data augmentation is not used in this paper on purpose in order to show that YOLO achieved good results with limited data.

3.2 Google Colab

All training and testing tasks are performed using a 12 GB NVIDIA Tesla K80 GPU provided by Google Colab which is compatible with DARKNET since, as mentioned earlier, DARKNET is written in C and CUDA. The NVIDIA CUDA deep neural network library (cuDNN) is used to make it all work.

3.3 Training and Testing

Since DARKNET-53 is pretrained, transfer learning is used to train YOLO on the NEU dataset. When training an object detector, it is always good to start from an existing model trained on very large datasets and then use the weights of this model to train. This is fine even if the trained weights do not contain the objects required in this experiment. This process is called transfer learning. A pretrained model that contains weights trained on ImageNet is used as starting weights so that the network can learn quicker. This is also beneficial since fewer data will be required [25] which is convenient since the NEU dataset only has 300 images per class before train/test split.

Several parameters are changed in order to train and test YOLOV3 using a custom dataset. However, one of the goals of this paper is to achieve this with very minor modifications to the network. Which is why most of the parameters are left the same way they came with YOLOV3. Some of the unchanged parameters include the loss function where YOLOV3 uses the sum-squared error in the loss function and to maximize the efficiency of this function, the network increases the confidence score as much as possible

for it to be equal to the IOU between the ground truth and the predicted bounding box and decreases the confidence score when there are no objects in the bounding box.

Another intact parameter is the activation function. YOLOV3 uses leaky activation function for each convolutional layer except the last one before each YOLO layer where a linear function is used. The linear activation function is also used in the shortcut connections.

At first, YOLO training reached an average loss of 0.11 which is supposed to be good; however, the network did not converge, the mAP was very low, no detections were made even on training images, true positive and false positive values were almost null and finally, the accuracy for each class was mostly 0.00% which meant more research and changes had to made in order to get better results (Fig. 3).

```
 calculation mAP (mean average precision)...
164
 detections_count = 55, unique_truth_count = 391
class_id = 0, name = Crazing, ap = 0.00%         (TP = 0, FP = 0)
class_id = 1, name = Inclusion, ap = 0.00%       (TP = 0, FP = 0)
class_id = 2, name = Patches, ap = 0.00%         (TP = 0, FP = 4)
class_id = 3, name = PittedSurface, ap = 0.00%           (TP = 0, FP = 2)
class_id = 4, name = RolledInScale, ap = 0.17%           (TP = 0, FP = 0)
class_id = 5, name = Scratches, ap = 0.00%       (TP = 0, FP = 1)

 for thresh = 0.25, precision = 0.00, recall = 0.00, F1-score = -nan
 for thresh = 0.25, TP = 0, FP = 7, FN = 391, average IoU = 0.00 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.000291, or 0.03 %
Total Detection Time: 11.000000 Seconds
```

Fig. 3. Results with 0.03% mAP

The next attempt was to troubleshoot the problem so 4 classes were removed, and then YOLO was left to train for only two classes which had defects easy to detect. After training, the results obtained were about the same; however, YOLO was able to detect defects in some images with very low confidence. But obviously, the results were not good considering only two classes were used.

Another trial was attempted where YOLO was left to train for more iterations and the results barely improved. This meant that the number of iterations was not the cause of bad results.

The dataset images were 200 × 200 in size; however, the network size was 416 × 416. YOLO has a built-in feature which allows it to resize images on its own in order to get the best out of the training; however, it was later discovered that this was not working properly since the network size was bigger than the images. After experimenting with resizing the images and resizing the network it was concluded that with a network size of 416 × 416 and image size of 608 × 608 the network achieved the best results so far (Fig. 4).

A Mini-batch gradient descent is used where a certain number of batches is taken during training. A Mini-batch gradient descent finds balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent. Smaller batch

```
 calculation mAP (mean average precision)...
164
 detections_count = 922, unique_truth_count = 391
class_id = 0, name = Crazing, ap = 0.24%          (TP = 0, FP = 0)
class_id = 1, name = Inclusion, ap = 25.88%       (TP = 14, FP = 5)
class_id = 2, name = Patches, ap = 36.71%         (TP = 29, FP = 21)
class_id = 3, name = PittedSurface, ap = 1.62%           (TP = 0, FP = 1)
class_id = 4, name = RolledInScale, ap = 9.11%           (TP = 4, FP = 2)
class_id = 5, name = Scratches, ap = 47.71%       (TP = 12, FP = 4)

 for thresh = 0.25, precision = 0.64, recall = 0.15, F1-score = 0.24
 for thresh = 0.25, TP = 59, FP = 33, FN = 332, average IoU = 46.04 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.202104, or 20.21 %
Total Detection Time: 11.000000 Seconds
```

Fig. 4. Results with 20.21% mAP

sizes are noisy, offering a regularizing effect and lower generalization error and make it easier to fit one batch worth of data in memory. The number of batches is lowered from 64 to 24 and the subdivisions from 64 to 8. The use of small batches as opposed to the typical use of large mini-batches, is proved to get better generalization and allows for a smaller memory footprint [26]. This is by far the most affecting factor in this experiment. The results improved significantly, and the network can converge and detect defects in all images including test images with high confidence.

4 Results and Analysis

In the final attempt, YOLO is trained for approximately 25000 iterations using the six types of defect images. As mentioned earlier, the batch number is lowered from 64 to 24 which helped raise the mAP. The learning rate is expected to start off high and then drop as the network learns and has more information and therefore requires less aggressive learning. This is exactly what happened; however at the beginning, the learning rate increased before reaching the point where it should decrease. This is called the burn-in period or the warmup period. Training took about 55 h on the single 12 GB NVIDIA Tesla GPU.

YOLO successfully made accurate detections and classifications on the test images provided with each detection taking up to an average of 85 ms. The network achieved a mAP of 70.66%, 79% precision and 68% recall. The results can be seen in Fig. 5. Some sample detections are shown in Fig. 7 where it can be seen how YOLO detects, localizes and classifies each of the six defects by drawing a bounding box around the defect and displaying the percentage of confidence as well as the time it took for detection.

As an example in Fig. 7, in the image containing patch defects, YOLO drew bounding boxes around what it thinks are patches. It is 100% confident that the defect is correctly classified, and it took only 88.77 ms for the whole process to be done.

The proposed model in this paper can also extract the coordinates of the resulting bounding boxes which in return allow obtaining the position and size of the defects. The network outputs the coordinates to a text file, along with the name and accuracy for each of the defects detected. Figure 6 shows detections on a metal sheet suffering

```
 calculation mAP (mean average precision)...
164
 detections_count = 848, unique_truth_count = 391
class_id = 0, name = Crazing, ap = 24.82%        (TP = 18, FP = 24)
class_id = 1, name = Inclusion, ap = 72.05%      (TP = 66, FP = 12)
class_id = 2, name = Patches, ap = 84.89%        (TP = 65, FP = 6)
class_id = 3, name = PittedSurface, ap = 87.79%          (TP = 32, FP = 3)
class_id = 4, name = RolledInScale, ap = 62.31%          (TP = 34, FP = 19)
class_id = 5, name = Scratches, ap = 92.09%      (TP = 50, FP = 5)

 for thresh = 0.25, precision = 0.79, recall = 0.68, F1-score = 0.73
 for thresh = 0.25, TP = 265, FP = 69, FN = 126, average IoU = 60.42 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.706573, or 70.66 %
```

Fig. 5. Final detection results

```
seen 64
Enter Image Path: /content/darknet/img/scratches_30.jpg: Predicted in 87.295000 milli-seconds.
Scratches: 83%      (left_x:  37 top_y:  10  width:  87  height: 205)
Scratches: 55%      (left_x: 184 top_y:  -3  width:  73  height:  84)
Scratches: 75%      (left_x: 258 top_y: 259  width:  82  height: 339)
Scratches: 53%      (left_x: 385 top_y:  34  width:  59  height: 273)
Scratches: 71%      (left_x: 391 top_y: 453  width:  46  height: 160)
Enter Image Path:
```

Fig. 6. Coordinate extraction

from many scratches with the network giving the accuracy for each scratch as well as the center coordinates, height and width of the bounding box enclosing the scratches. The model can also make predictions in a matter of milliseconds and can be deployed on mobile devices such as cameras to be used in production lines since it is considered lightweight and can perform fast detections on just about any regular laptop. It can smoothly track and detect defects and it is robust enough when it comes to changes in size and orientation.

Although YOLO can make detections and classifications correctly, one of the defects, crazing, has an average precision (AP) of 24% as opposed to the high AP that the other defects have. This is return caused the mAP to drop. Even with a lower AP, the network is still able to detect and classify crazing defects and with high accuracy.

The other papers mentioned in the background section, used either the NEU dataset [17] or preferred to gather their own images and labels or use known datasets [16, 18, 19] which are very similar to the NEU dataset images. Another thing to note is that a direct comparison of this paper with the other methods is not possible since most methods used have their own metric for measuring performance depending on the model used. In YOLOs case, the mAP is used. When it comes to detectors such as YOLO it is much better to use the mAP metric instead. The mAP has many advantages over other metrics, like avoiding the "accuracy paradox" which is when accuracy increases even though the model is not actually good. This usually happens when True Positive (TP) < (False Positive) FP.

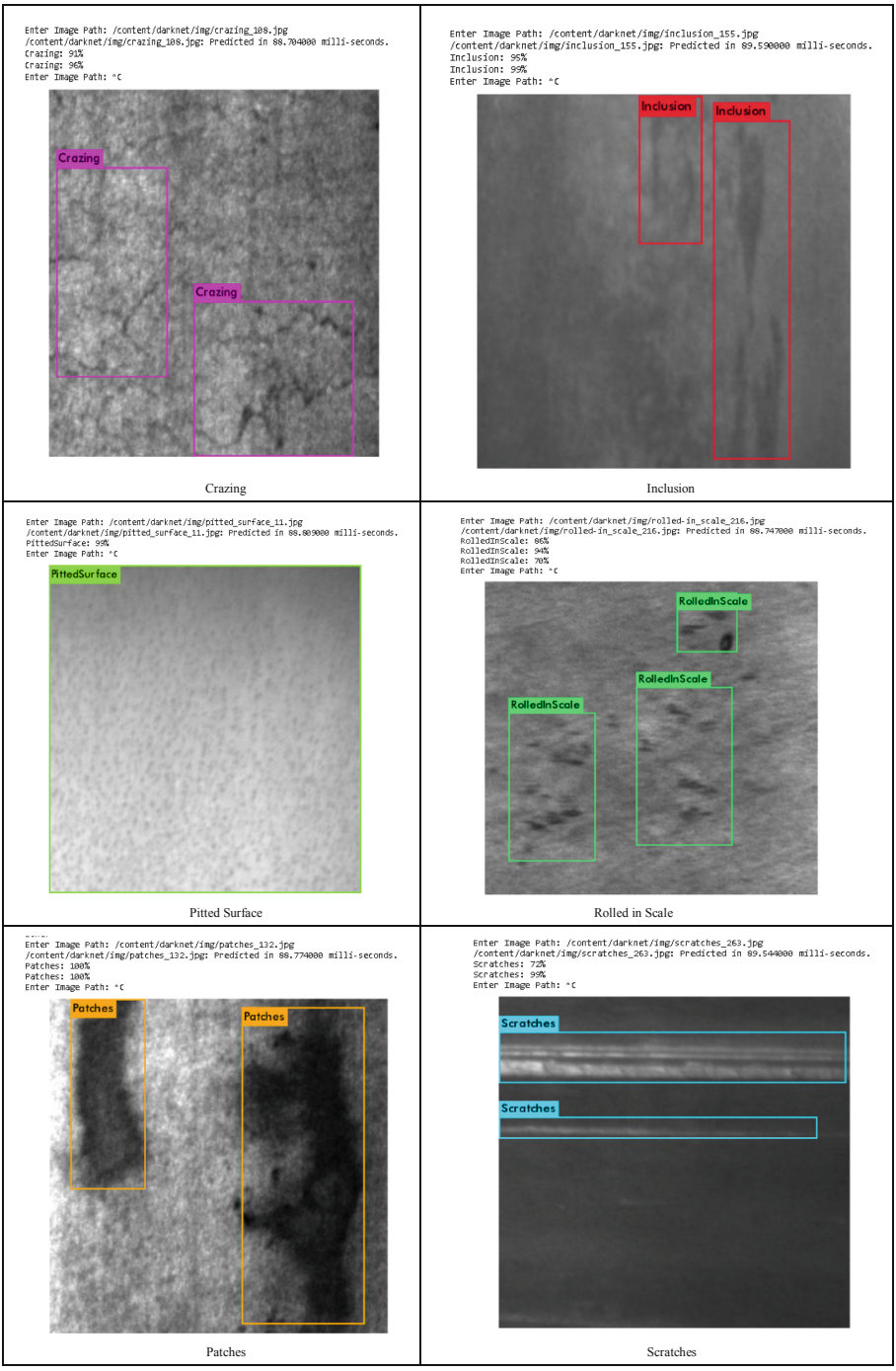


Fig. 7. Sample detections

5 Conclusion

YOLOV3 detector is modified and then trained on a dataset containing six types of defects on steel surfaces. The dataset is prepared and the labels configured to fit the YOLO format. After many trials and changes to the hyperparameters such as batch size and network size, YOLO is able to achieve a mAP of 70.66% with 79% precision and 68% recall. Most of the defects have high average precision with the exception of one which received 24% which in return affected the mAP. This is a limitation which might be overcome by applying noise reducing filters. Despite this, the network still achieves accurate detections and classifications taking up to an average of 85 ms. It must be noted as well that the results in this paper are obtained using a relatively small dataset with no data augmentation which is usually not enough to train a neural network or achieve decent results. The network also obtains the coordinates of resulting bounding boxes in order to calculate the sizes and locations of the defects. This is important for improving the manufacturing process and the quality of products rolled out of factories. It is important to note that even though YOLO trained on metal steel surfaces, it can be used and trained on other surfaces such as wood, glass and paper.

Further work includes heavier modifications to the source files and hyperparameters such as learning rate, anchors, loss function and even altering the layers of the network by changing the values of filters and maybe adding or removing certain layers. Accuracy may also be improved in the case of a bigger dataset, preprocessing of the data and data augmentation techniques.

References

1. Sun, Q., Cai, J., Sun, Z.: Detection of surface defects on steel strips based on singular value decomposition of digital image. Math. Probl. Eng. 1–12 (2016)
2. Jiang, Q., Sun, C., Liu, X., Hong, Y.: Very-high-cycle fatigue behavior of a structural steel with and without induced surface defects. Int. J. Fatigue **93**, 352362 (2016)
3. Neethu, N.J., Anoop, B.: Role of computer vision in automatic inspection systems. Int. J. Comput. Appl. **123**(13), 28–31 (2015)
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440 (2015)
5. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
6. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
7. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S.: Deep residual learning for image recognition, pp. 770–778 (2016)
8. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G.: Steel defect classification with max-pooling convolutional neural networks. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–6 (2012)
9. Soukup, D., Huber, R.: Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images. ISVC (2014)
10. Weimer, D., Scholz-Reiter, B., Shpitalni, M.: Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Ann. **65**(1), 417–420 (2016)

11. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv:180402767 (2018)
12. Pyimagesearch Intersection over Union (IoU) for object detection. https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-forobject-detection/. Accessed 24 Jan 2020
13. Redmon, J.: You Only Look Once: Unified, Real-Time Object Detection. Las Vegas, NV (2016)
14. Redmon, J.: Darknet: Open Source Neural Networks in C. Pjreddie.com (2019)
15. GitHub AlexeyAB/darknet. https://github.com/AlexeyAB/darknet. Accessed 24 Jan 2020
16. Zhou, S., Chen, Y., Zhang, D., Xie, J., Zhou, Y.: Classification of surface defects on steel sheet using convolutional neural networks. Materiali in tehnologije **51**(1), 123–131 (2017)
17. Islam, F., Rahman, M.: Metal Surface Defect Inspection through Deep Neural Network (2018)
18. Li, J., Su, Z., Geng, J., Yin, Y.: Real-time detection of steel strip surface defects based on improved YOLO detection network. IFAC-PapersOnLine **51**(21), 76–81 (2018)
19. Tabernik, D., Šela, S., Skvarč, J., Skočaj, D.: Segmentation-based deep-learning approach for surface-defect detection. J. Intell. Manuf. **31**(3), 759–776 (2019). https://doi.org/10.1007/s10845-019-01476-x
20. Song, K., Yan, Y.: A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl. Surf. Sci. **285**, 858–864 (2013)
21. He, Y., Song, K., Meng, Q., Yan, Y.: An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. IEEE Trans. Instrum. Meas. **69**(4), 1493–1504 (2019)
22. He, Y., Song, K., Dong, H., Yan, Y.: Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. Opt. Lasers Eng. **122**, 294–302 (2019)
23. Bulkresizephotos. Bulkresizephotos.com. Accessed 24 Jan 2020
24. WINTRISS INSPECTION SOLUTIONS surface inspection. http://www.winspection.com/surface-inspection.php. Accessed 24 Jan 2020
25. Aytar, Y., Zisserman, A.: Tabula rasa: model transfer for object category detection. In: 2011 International Conference on Computer Vision, pp. 2252–2259 (2011)
26. Masters, D., Luschi, C.: Revisiting small batch training for deep neural networks. arXiv:180407612 (2018)

Methods of the Vehicle Re-identification

Mohamed Nafzi[1(✉)], Michael Brauckmann[1], and Tobias Glasmachers[2]

[1] Facial & Video Analytics, IDEMIA Identity & Security Germany AG, Bochum, Germany
{mohamed.nafzi,michael.brauckmann}@idemia.com
[2] Institute for Neural Computation, Ruhr-University Bochum, Bochum, Germany
tobias.glasmachers@ini.rub.de

Abstract. Most of researchers use the vehicle re-identification based on classification. This always requires an update with the new vehicle models in the market. In this paper, two types of vehicle re-identification will be presented. First, the standard method, which needs an image from the search vehicle. It produces a feature vector, which will be applied by the re-identification of the search vehicle. VRIC and VehicleID data set are suitable for training this module. It will be explained in detail how to improve the performance of this method using a trained network, which is designed for the classification. The second method takes as input a representative image of the search vehicle with similar make/model, released year and colour. It is very useful when an image from the search vehicle is not available. It produces as output a shape and a colour features. This could be used by the matching across a database to re-identify vehicles, which look similar to the search vehicle. To get a robust module for the re-identification, a fine-grained classification has been trained, which its class consists of four elements: the make of a vehicle refers to the vehicle's manufacturer, e.g. Mercedes-Benz, the model of a vehicle refers to type of model within that manufacturer's portfolio, e.g. C Class, the year refers to the iteration of the model, which may receive progressive alterations and upgrades by its manufacturer and the perspective of the vehicle. Thus, all four elements describe the vehicle at increasing degree of specificity. The aim of the vehicle shape classification is to classify the combination of these four elements. The colour classification has been separately trained. After the training, the classification layer will not be used. By both methods, even data of vehicles by some makes/models/released years/perspectives or by some colours are not available, it will be possible to re-identify each vehicle. The results of vehicle re-identification will be shown. Using a developed tool, the re-identification of vehicles on video images and on controlled data set using a search image will be demonstrated. The results of a proposed mix-mode, which is the combination of shape matching and colour classification, will be presented. This work was partially funded under the grant.

Keywords: Vehicle re-identification · Mix-mode · CNN · Shape and colour classification

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 516–527, 2021.
https://doi.org/10.1007/978-3-030-55180-3_38

1 Introduction

The objective of the vehicle re-identification module is to recognize a vehicle within a large image or video data set. Two different methods will be trained and tested.

- First, the standard vehicle re-identification. The known data set VRIC and VehicleID have been used separately for training and testing. VRIC data set contains 2811 vehicle-IDs with 54808 images and VehicleID contains 13164 vehicle-IDs with 113346 images for training. Also Multiple loss and a merged data set have been used to train on both data set. This, can increase the robustness of the module. Starting the training from the scratch using a network, which has been trained on shape classification using about eight million images, can significantly improve the results. The results of the fusion will be also presented.
- In the training of the second method, which requires just a representative image looks similar to the search vehicle in case its sample image is not available, a fine-grained vehicle classification has been used, which leads to feature representation with small intra-class variance. The modules have been trained using CNN-Networks. The combination of the shape and the colour feature vectors leads to a robust re-identification of vehicles.
  - Training: Typically, a fine-grained class consists of four elements: the make of a vehicle refers to the vehicle's manufacturer, e.g. Mercedes-Benz, the model of a vehicle refers to type of model within that manufacturer's portfolio, e.g. C Class, the year refers to the iteration of the model, which may receive progressive alterations and upgrades by its manufacturer and the perspective of the vehicle. Thus, all four elements describe the vehicle at increasing degree of specificity. The aim of the vehicle shape classification is to classify the combination of these four elements. We trained our vehicle shape network on 11906 classes using about eight million images. We trained the colour classification separately on 10 classes using about two million images.
  - Application: In the application of our trained CNN-Network, the classification layer will not be used. Our module supports searches using an image sample or a representative image of the search vehicle, which is sent to the template creation component. The search engine performs the template matching across a video database using shape and colour features and returns the search results to the user. This method does not require the training of all vehicle classes. To get an alarm the make, the model, the released year, the perspective and the colour of the probe and of the gallery images should be similar.

2 Related Works

Some research has been performed on make/model classification to re-identify a search vehicle. Most of it operated on a small number of make/models because

it is difficult to get a labeled data set panning all existing make/models. Manual annotation is almost impossible because one needs an expert for each make being able to recognize all its models and it is very tedious and time consuming process. Author in [9] developed a make/model classification based on feature representation for rigid structure recognition using 77 different classes. Two distances have been tested, the dot product and the euclidean distance. Author in [7] tested different methods by make/model classification of 86 different classes on images with side view. The best one was HoG-RBF-SVM. Author in [10] used 3D-boxes of the image with its rasterized low-resolution shape and information about the 3D vehicle orientation as CNN-input to classify 126 different make/models. The module of [8] is based on 3D object representations using linear SVM classifiers and trained on 196 classes. In a real video scene all existing make/models could occur. Considering that we have worldwide more than 2000 models, make/model classification trained just on few classes will not succeed in practical applications. Author in [6] increase the number of the trained classes. His module is based on CNN and trained on 59 different vehicle makes as well as on 818 different models. His solution seems to be closer for commercial use. Our developed module in our previous work [1] was trained on 1447 different classes and could recognize 137 different vehicle makes as well as 1447 different models of the released year between 2016 till 2018. Other research has been operated on the known standard vehicle re-identification. Space-time contextual knowledge has been exploited for vehicle re-id subject to structured scenes. Author in [3] incorporated spatio-temporal path information of vehicles. This method improves the re-id performance on the VeRi-776 data set, it may not generalize to complex scene structures when the number of visual spatio-temporal path proposals is very large with only weak contextual knowledge available to facilitate model decision. [4] considered 20 vehicle key points for learning and aligning local regions of a vehicle for re-identification. Clearly, this approach comes with extra cost of exhaustively labeling these key points in a large number of vehicle images, and the implicit assumption of having sufficient image resolution/details for computing these key points. Author in [5] worked on VehicleID data set, which includes multiple images of the same vehicle captured by different real world cameras in a city. This data set is challenging in term to separate between similar vehicles with few of differences but it is only constrained test scenarios due to the rather artificial assumption of having high quality images of constant resolution. This makes them limited for testing the true robustness of re-id matching algorithms in typically unconstrained wide-view traffic scene imaging conditions. Author in [2] introduced the Veric data set to address the limitation of other Vehicle re-identification Benchmarks, which provides conditions giving rise to changes in resolution, motion blur, weather, illumination, and occlusion. In this paper, we show two methods of the vehicle re-identification, which could re-identify vehicles even if their classes are not included in the training. First method is the standard vehicle re-identification, which requires a probe image of the search vehicle. This module has been trained using a merged data set of Veric and VehicleID. Its training has been started from the scratch of the

make/model network used in the second method, which is trained on classification using 11906 classes with about eight million images for the shape and using 10 classes with about two million images for the colour. It uses shape and colour feature vectors for the re-id. It works even if a probe image of the search vehicle is not available. A representative image with similar make, model, released year and colour of the search vehicle would be enough for the re-identification. It could be downloaded e.g. from the web. Experimental results show that the first method outperforms all state-of-the-art approaches on Veric and VehicleID data set. Here, the comparison has been done just to the best published results. The second method helps to improve the performance of the first method, and it gives a solution in case a probe image of the search vehicle is not available. Here, there are no defined data set we could use to compare the results to other research. Tests has been evaluated on an internally data set.

3 Network Architectures and Feature Extraction

Neural networks have been used in computer vision for a long time, but with the progress in hardware capabilities and growth of available training data over the last few years deep neural networks have become the most successful methods for many computer vision tasks. In some visual recognition tasks, even human-level accuracy can be surpassed. We used a CNN-networks based on ResNet architecture. Their coding time is 20 ms (CPU 1 core, i7-4790, 3.6 GHz). In Fig. 1 and 2, we show our way to extract the feature vector, which will be used in the matching step by the vehicle re-identification. The Fig. 1 shows the trained CNN for the vehicle re-identification based on shape and colour classification (method 2), and Fig. 2 shows the trained CNN for the standard vehicle re-identification (trained on gray images/method 1). Here, we started from the scratch using the CNN from the method 2, which has been trained on 11906 classes with about eight million images. This CNN-net is an expert to separate between vehicles with different makes, models or released years. By this way the training is focusing to separate between different vehicles with similar makes, models and released years but without forgetting to separate between vehicles with different makes, models or released years. Here, two CNN-nets have been trained. By the first training all parameters are trainable. By the second CNN-net the convolution block is not trainable. Here, the training tunes just the IP-Layer for the separation between the classes. The fusion shows the best results on Veric and VehicleID.

4 Matching and Fusion

Our feature vectors (templates) are normalized to unit length. The matching as such is performed by calculating the dot product between two feature vectors which i.e. the cosine of the angle between both vectors. Hereby by the method 2, the matching scores of the color and of the shape feature vectors have different distributions. Fusion uses a weighted sum of the match scores of both modalities.

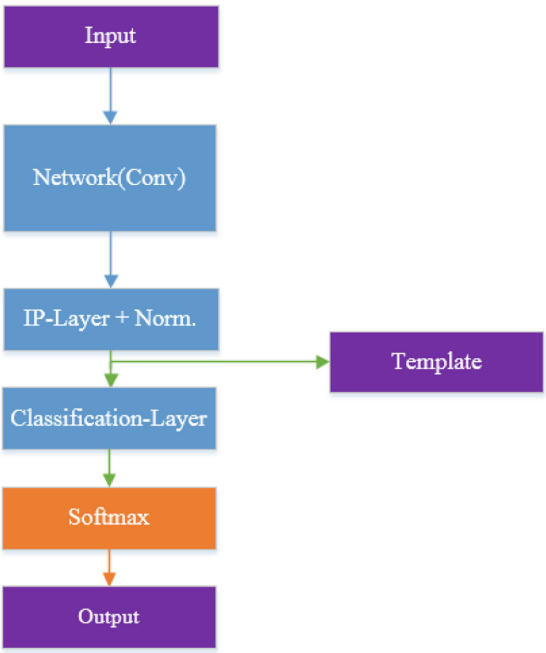


Fig. 1. Feature vector extraction. The network CNN1 for the vehicle re-identification based on shape and colour classification (method 2). Trained on 11906 classes for shape and on 10 classes for colour.

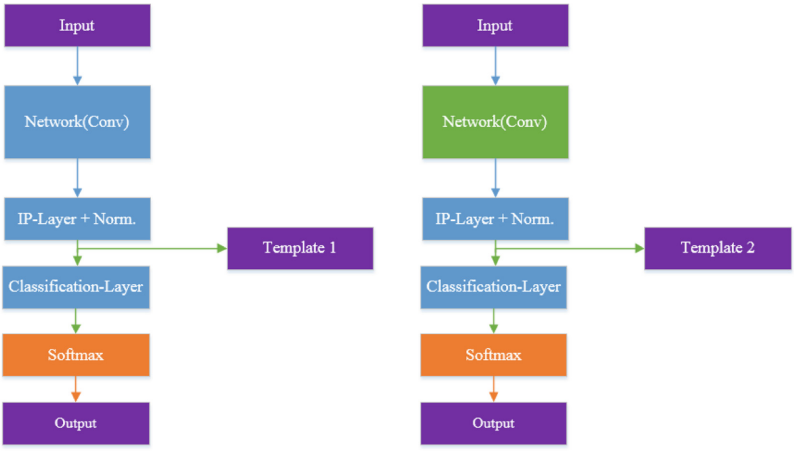


Fig. 2. Feature vector extraction. The network CNN2 for the standard vehicle re-identification (method 1). Starting from the scratch (using trained CNN1 from the method 2). Blue indicates trainable parameters. Green shows not trainable parameters. Both CNNs are trained on a merged data set of Veric and VehicleID. CNN2 is the fusion of CNN-nets and shows the best results.

Optimal weights have been determined based on a predefined set of data. By method 1, the fusion score is the sum of the match scores, which have similar distributions.

5 Mix-Mode

According to our method 2 for vehicle re-identification based on shape and colour features, we need for a vehicle search a respective search image of a certain make/model, released year and color. The make and model of the search image does not need to be part of the make/models categories used during training. In practice, we could have the case that we have an image just with the same shape but not with the same color of the search vehicle, e.g. downloaded from a manufacturer's internet homepage. In this case, we could apply our developed Mixed-Mode, which is the appropriate solution for this problem. In this mode, we combine the shape matching together with color classification. We use the shape feature vector for matching. As results, we get all vehicles that have the same shape as the searched vehicle however potentially with different colors. After that, we apply the color classification to filter the results by the selected color. This mode is intended specifically to be used in investigational scenarios.

6 Experiments

6.1 Experiments of the Vehicle Re-identification Based on Shape and Colour Features

In total, 406 best-shots and 85.130 detections were computed from Cam2, and 621 best-shots with 199.963 detections from Cam4. Additionally, 33 controlled images were acquired from the web ("Google") for subsequent experiments. Based on these VICTORIA data sets, we performed a number of tests using

Fig. 3. The image on the left side shows a sample of a best-shot computed from the VICTORIA data set ("Cam2"). The image on the right side depicts a best-shot from "Cam4", respectively.

Fig. 4. The color "silver" is not included in our training of color classification. Right vehicle is labeled as gray but with sunlight looks close to white. It produces higher impostor scores with white vehicles like the vehicle on the left, this leads to a reduction of the verification rate as depicted by the black ROC curves in Fig. 5 and 6

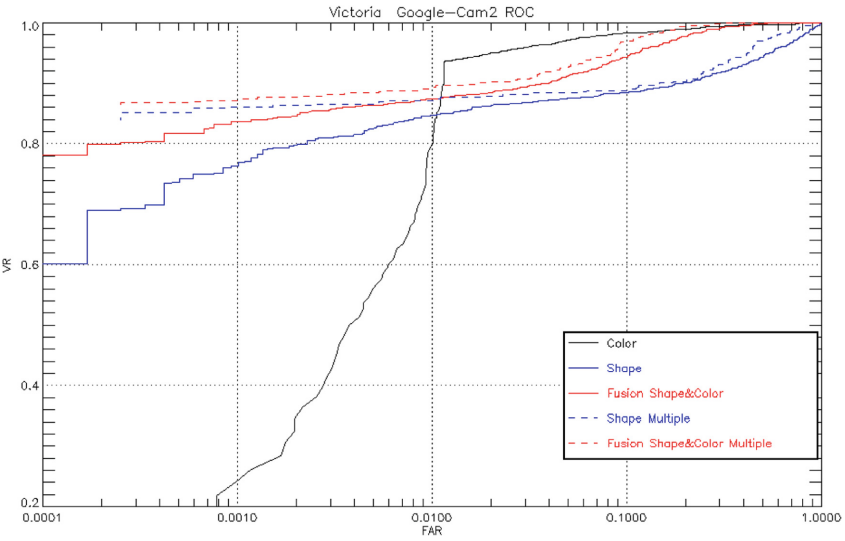


Fig. 5. This figure shows ROC-curves of shape, color, fusion of color and shape and using multiple probe images by shape. Computation was done matching of controlled single images from the internet against video data set Cam2 from the project Victoria. **Color**: matching using color template (black curve). **Shape**: matching using shape template (blue solid curve). **Fusion Shape&Color**: Fusion of shape and color matching scores (red solid curve). **Shape Multiple**: matching using shape template and using multiple probe images (blue dashed curve). **Fusion Shape&Color Multiple**: Fusion of shape using multiple probe images and color matching scores (dashed solid curve). **FAR**: False Acceptance Rate. **VR**: Verification Rate.

the shape feature, the colour feature and the fusion of both. multiple probe images by shape matching have been also tested. Here, we have a set of images of the search vehicle with different views. By matching across a gallery image, we get a set of scores. Their maximum is the finale match score. This reduces the dependency of the perspective by matching. Tests have been evaluated on video data across still images. The Fig. 3 shows sample images from the video data set Cam2 and Cam4. Results are shown in the Fig. 5 and 6. Here as shown, we got some high impostor scores by matching of color templates, leading to a fall of the ROC curves. The reason for this is that the color "silver" is currently not included in the classes used for the training, thus we labelled it as "grey". Due to the sun-light conditions however, the silver color was mapped onto "white". The Fig. 4 shows two sample images illustrating this effect.

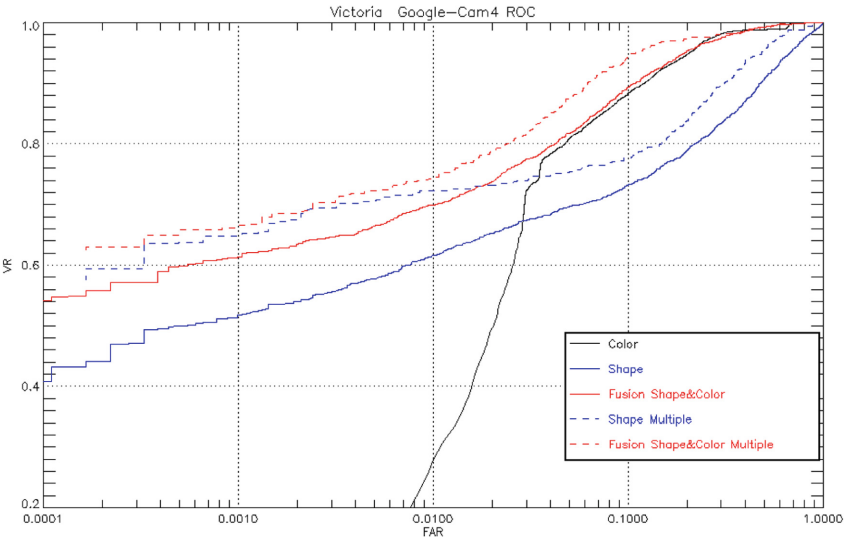


Fig. 6. This figure shows ROC-curves of shape, color, fusion of color and shape and using multiple probe images by shape. Computation was done matching of controlled single images from the internet against video data set Cam4 from the project Victoria. **Color**: matching using color template (black curve). **Shape**: matching using shape template (blue solid curve). **Fusion Shape&Color**: Fusion of shape and color matching scores (red solid curve). **Shape Multiple**: matching using shape template and using multiple probe images (blue dashed curve). **Fusion Shape&Color Multiple**: Fusion of shape using multiple probe images and color matching scores (dashed solid curve). **FAR**: False Acceptance Rate. **VR**: Verification Rate.

6.2 Experiments of the Standard Vehicle Re-Identification

Data Sets. For evaluation, we utilised two most popular vehicle re-identification benchmarks. The VehicleID data set [5] provides a training set with 113,346

images from 13,164 IDs and a test set with 17,377 probe images and 2,400 gallery images from 2,400 identities. It adopts the single-shot re-id setting, with only one true matching for each probe. The VRIC data set [2] has 54,808 images from 2,811 IDs in training set. The probe and the gallery of the testing data set contain 2,811 images with 2,811 vehicle IDs. The data split statistics are summarised in Table 1. The Fig. 7 shows a hard negative pair.

Evaluation. Table 2 compares our method1 (CNN2) explained in sections before with state-of-the-art methods on two benchmarks. Our method outperforms all other competitors with large margins. It surpasses the best competitor in Rank-1 rate by 8.53% (this means 16.0% error reduction) and in Rank-5 by 9.55% on VRIC, and in Rank-1 rate by 2.8% (this means 7.6% error reduction) and in Rank-5 by 4.2% on VehicleID.

Table 1. Data split of standard vehicle re-identification data sets evaluated in our experiments.

Dataset	Training IDs/Images	Probe IDs/Images	Gallery IDs/Images
VehicleID [5]	13,164/113,346	2,400/17,377	2,400/2,400
VRIC [2]	2,811/54,808	2,811/2,811	2,811/2,811

Table 2. Comparative of standard vehicle re-identification results on two benchmarking data sets.

Method	VehicleID [5]		VRIC [2]	
	Rank-1 in %	Rank-5 in %	Rank-1 in %	Rank-5 in %
OIFE (Single Branch) [4]	32.86	52.75	24.62	50.98
Siamese-Visual [3]	36.83	57.97	30.55	57.30
MSVF [2]	63.02	73.05	46.61	65.58
Our method 1 (CNN2)	65.82	77.25	55.14	75.13

7 Manual Testing Using Our Vehicle Re-identification Tool

Besides the statistical experiments from the section before, we performed manual tests on the second method trained on shape and colour features with the vehicle re-identification tool. We tested also the mix-mode, which has been defined in this research. The Fig. 8 shows exemplary the search for a green "Ford Ka". The left side of the figure depicts the selected search image, the middle part shows the best-shots of the matches against the VICTORIA data (Cam3 video sequence), and the right side presents all detections belonging to the selected best-shot. The subsequent Fig. 9 shows an example for the Mixed Mode. In this

Fig. 7. Example of hard negative pair.

Fig. 8. This figure shows the vehicle re-identification based on shape and color features.

scenario, the user searches for a 'white' Hummer 2. In case that a sample image of that Hummer 2 is available, however with a different color, here 'orange', he nevertheless can apply the search that provides all occurrences of that Hummer 2 however with any color. In a follow-up step, color classification is applied to filter those result images with the searched color, here 'white'.

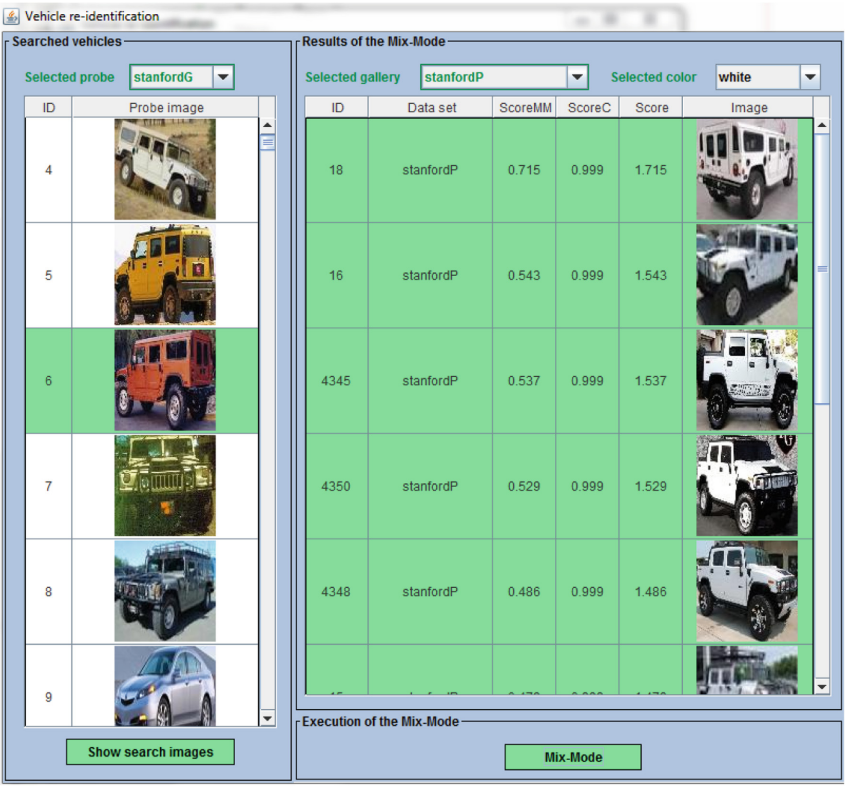


Fig. 9. This figure shows the vehicle re-identification using the Mix-Mode based on shape feature and color classification.

8 Conclusion and Future Work

- Both vehicle re-identification methods work on classes even if they are not included in the training. They have not immediately to be updated with new released models.
- The perspectives of the probe and of the gallery samples by mates should be similar to get an alarm. Using multiple probe images with different views make the re-identification independently of the perspective.
- Vehicle re-identification based on shape and colour classification works even if an image of the search vehicle is not available. A representative image is sufficient. It re-identifies all vehicles with similar makes, models, released years and colours.
- An image of the search vehicle is required for the standard re-identification, which could re-identify exactly the same vehicle.
- The training of the Vehicle re-identification based on shape classification helps the training of the standard re-identification because the size of the training

data of the first training is much larger than the second training. Its results beats the best published methods as shown in the Table 2.
- We are working on the classification of the perspective of the vehicle based on image or template.
- We plan to augment training data for the standard vehicle re-identification.
- We are working on different methods to improve the vehicle shape classification.

Acknowledgment. –Victoria: funded by the European Commission (H2020), Grant Agreement number 740754 and is for Video analysis for Investigation of Criminal and Terrorist Activities.

–Florida: funded by the German Ministry of Education and Research (BMBF).

References

1. Nafzi, M., Brauckmann, M., Glasmachers, T.: Vehicle shape and color classification using convolutional neural network. CoRR, abs/1905.08612, March 2019. http://arxiv.org/abs/1905.08612
2. Kanaci, A., Zhu, X., Gong, S.: Vehicle re-identification in context. CoRR, abs/1809.09409, October 2018. http://arxiv.org/abs/1809.09409
3. Shen, Y., Xiao, T., Li, H., Yi, S., Wang, X.: Learning deep neural networks for vehicle re-ID with visual-spatio-temporal path proposals. CoRR, abs/1708.03918, August 2017. http://arxiv.org/abs/1708.03918
4. Wang, Z., Tang, L., Liu, X., Yao, Z., Yi, S., Shao, J., Yan, J., Wang, S., Li, H., Wang, X.: Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017). https://doi.org/10.1109/ICCV.2017.49
5. Liu, H., Tian, Y., Wang, Y., Pang, L., Huang, T.: Deep relative distance learning: tell the difference between similar vehicles. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 2167–2175 (2016). http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7780607&isnumber=7780329
6. Dehghan, A., Masood, S.Z., Shu, G., Ortiz, E.G.: View independent vehicle make, model and color recognition using convolutional neural network. CoRR, abs/1702.01721 (2017). http://arxiv.org/abs/1702.01721
7. Boyle, J., Ferryman, J.: Vehicle subtype, make and model classification from side profile video. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6, August 2015. https://doi.org/10.1109/AVSS.2015.7301783
8. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: 2013 IEEE International Conference on Computer Vision Workshops, pp. 554–561, December 2013. https://doi.org/10.1109/ICCVW.2013.77
9. Petrovic, V., Cootes, T.F.: Analysis of features for rigid structure vehicle type recognition. In: Proceedings of the British Machine Vision Conference, United Kingdom, vol. 2. BMVA (2004)
10. Sochor, J., Herout, A., Havel, J.: Boxcars: 3D boxes as CNN input for improved fine-grained vehicle recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3006–3015 (2016)

A Novel Cognitive Computing Technique Using Convolutional Networks for Automating the Criminal Investigation Process in Policing

Francesco Schiliro[1(✉)], Amin Beheshti[1], and Nour Moustafa[2]

[1] Department of Computing, Macquarie University, Sydney, Australia
francesco.schiliro@hdr.mq.edu.au, amin.beheshti@mq.edu.au
[2] University of New South Wales Canberra @ Adfa, Canberra, Australia
nour.moustafa@unsw.edu.au

Abstract. Criminal Investigation (CI) plays an important role in policing, where police use various traditional techniques to investigate criminal activities such as robbery and assault. However, the techniques should hybrid with the use of artificial intelligence to analyze and determine different crime types for taking actions in real-time. In contrast with the manual process of investigating a large amount of data collected related to a criminal investigation. In this paper, we present a novel Cognitive Computing enabled Convolution Neural Networks (CC-CNN) approach for identifying crime types, such as robbery and assault, collected from unstructured textual data. We develop learning algorithms and provide a cognitive assistant to assist a police investigator in easily understanding crime types. We train and validate the CC-CNN technique on two datasets including handcrafted text-crime dataset and sentiment polarity dataset of negative and positive reviews. The experimental results show that our approach performs at a high level in terms of accuracy, error rate and time processing using both datasets.

Keywords: Crime Investigation · Convolution Neural Networks · Cognitive Computing · Police investigation

1 Introduction

Serious crimes have become ongoing criminal justice threats, which considerably contributes to the risks with high costs nationally and globally [1]. Police have the full responsibility to investigate, react and respond to serious crimes using a criminal investigation process. This process aims to legally investigate the crime scene to lawfully collect evidence, concerning the rights of victims. It includes the powers of police to keep accused, interrogate and to find out and seize property, across the suitable use of warrants and other legal procedures [2]. Police investigators usually utilize various methods and techniques to investigate crimes that have a great impact on examining whether a lawbreaker is defined, arrested and/or makes a confession, which can affect either case are well-defined or convictions secured [1].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 528–539, 2021.
https://doi.org/10.1007/978-3-030-55180-3_39

Criminal Investigation (CI) is a complex and step-wise problem-solving process that involves the employment of a diversity of investigative strategies and making decisions at every phase to ascertain whether an individual charged with wrongdoing is guilty, or if a suspect should be prosecuted for an offense [19]. The commencement of a criminal inquiry can be based on reactive responses to general public reports, recommendations by other investigative agencies, identification of criminal patterns from ongoing inquiries, or re-investigations following the emergence of novel information. Similarly, law enforcement officers can embrace intelligence-gathering technologies to proactively investigate ongoing illegal practices or prevent looming delinquency [20]. After arriving at the crime scene, a police officer should quickly screen the environment and make critical choices, often grounded on limited data within a dynamic setting of active and ongoing actions [21]. Once a felonious event ceases to be active, the officer is anticipated to preserve the scene, gather evidence, and formulate a fact-finding plan that will result in the establishment of reasonable grounds for identifying and charging the perpetrators of the delinquency.

To illustrate the police investigation of crimes, the process starts when a victim reports a crime, such as a robbery and assault, to the police department. During the investigation process, the police officer may request statements of the crime description from victims, statements from witnesses if possible, and a collection of evidence to the crime [3]. The victims and witnesses make statements to police, which include detailed information about describing the crime and answering questions that provide more details about what happened. The collected information provides significant evidence, which assists police charge the offenders and prevent them from committing the crime again [31–33]. The police gather all the evidence of the crime scene, such as photographs, clothes and fingerprints, to present them at court [4]. The process of investigating crimes using data analytics is still manual, which needs specialist investigators to determine cases and crime types [1, 3]. The manual investigation is ineffective and takes a long time to examine a huge amount of data collected about crime and find significant evidence. Ineffective investigations of crimes can have consequences on many levels for victims, the general public, police and the criminal justice system. There is a threat that further major crimes could be committed, and victims could be reluctant to report serious crimes in the future [1, 5]. An ineffective manual investigation process could be automated by inspecting data collections of crimes using Artificial Intelligence to find intelligent patterns and identify evidence of crimes [29, 30].

In this study, we present a novel Cognitive Computing enabled Convolution Neural Networks (CC-CNN) to classify crime categories and their criminal acts. We develop learning algorithms and provide a cognitive assistant to assist a police investigator in easily understanding crime types. The cognitive assistant will enable predicting crimes by analyzing victim and witness statements, and any annotations and notes entered by the police investigator. We train and validate the CC-CNN technique on two datasets including handcrafted text-crime dataset (includes 10 crime types, e.g., sacrilege and housebreaking, larceny and credit stealing, and over 50000 criminal cases) and sentiment polarity dataset of negative and positive reviews. The experimental results show that our approach performs at a high level in terms of accuracy, error rate and time processing using both datasets. The remainder of this paper is structured as fellows. Section 2

explains the background and related studies on investigating crimes using CNN. The proposed methodology is discussed in Sect. 3. In Sect. 4, we present the experimental study before concluding the paper with remarks for future directions in Sect. 5.

2 Background and Related Studies

This section describes the concepts related to the crime investigation process and the use of deep learning algorithms for investigating crime types.

2.1 Crime Investigation Process

A crime investigation process is a strategy applied by police to collect evidence to define and arrest criminals, elicit confessions, close cases and secure [6]. A criminal investigation can be instigated using either a reactive (e.g., a report from the general public or referral by other agencies) or a proactive (e.g., a crime pattern analysis or operational intelligence assessment) approach [7]. The police have a policy that guides call takers, public counter staff and patrol officers on the information that they need to gather and subsequent action to take. Most crimes reported to the police are not major incidents and usually, the officer who first attends is the only resource that is required. This officer could be the investigator throughout the inquiry.

The quality of an initial investigation of a crime, whether carried out in person or over the telephone, is a significant factor in gathering material that leads to the detection of a crime, as depicted in Fig. 1. There may be limited opportunities to locate and gather material and those who conduct the initial investigation must ensure that material is not lost. The initial investigation phase is concluded when several actions have been completed. A crime is allocated for further investigation, investigators should develop a clear plan for how they intend to bring the investigation to a successful conclusion. Despite criminal investigation is a considerable role of police for dealing with serious crimes, obtaining evidence often lacks the level of synthesis seen in other fields of policing demand multiple steps of investigations [1, 7]. The use of deep learning algorithms, specifically CNN, could help to automatically learn and infer evidence for problems of crime scene investigation [8].

The CI process is broadly described as the police endeavour to gather evidence that will result in the identification and capture of the committer of an offense, and that will ensure the prosecutor obtains a conviction [20, 21]. Baber et al. [22] define CI as a sequential procedure that commences with a report of lawbreaking and concludes with the apprehension of a suspect or the exhaustion of all lines of investigations and the filing of the crime. This work outlines eight steps in the CI process which are equivalent to ones from the College of Policing's [20] instigation, including initial investigation, investigative evaluation, additional inquiry, suspect management, evidential evaluation, charge, and court presentation phases.

Braga et al. [23] highlight the significance of the fast-track actions indicating that the most serious criminalities are resolved by the first responders, often patrol officers, through information gathered from the victim(s) instead of leads generated by criminal

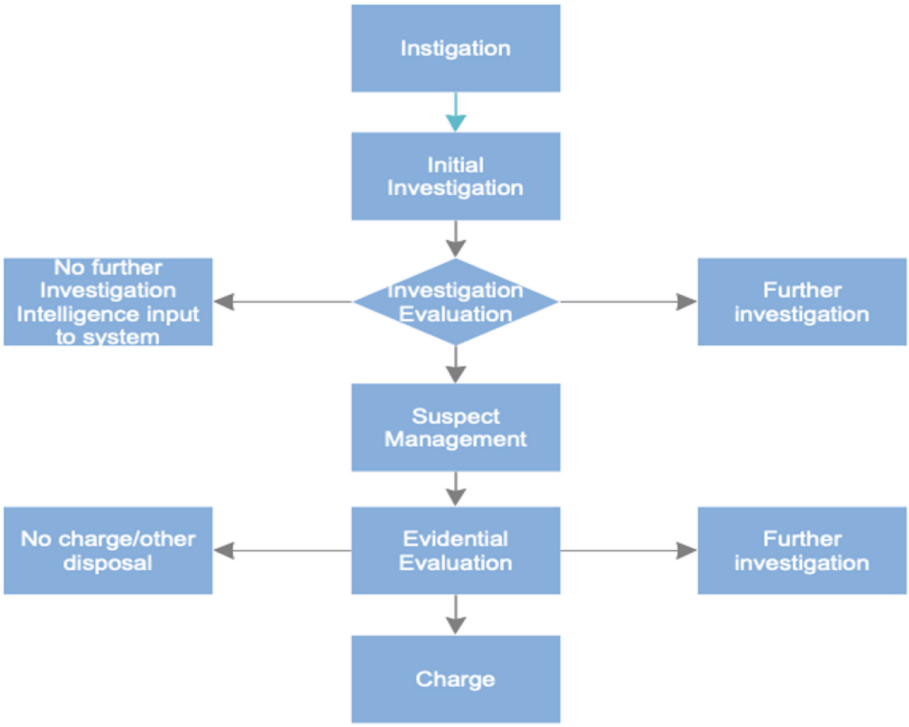


Fig. 1. Investigation process for defining criminal events.

investigators. This argument is grounded on the landmark outcomes of a Rand Corporation publication in 1975, which found that in more than 50% of the solved cases, the perpetrator's identity is established or quickly determined at the time the crime was reported in [23]. The College of Policing [24] recommends that police officers should carry out comprehensive documentation of all the occurrences in the initial investigation phase. The latter eases the exploratory evaluation, contributes to the creation of an intelligence snapshot of the crime surrounding, permits supervisors to review the quality of the inquiry, and facilitates the abdication of the investigation in case it is assigned to another criminal detective.

The concluding stage of the initial investigation is followed by the analytical evaluation phase, where the inspector evaluates the gathered data and establishes whether there is a need for additional lines of inquiries [22]. Whenever a crime file necessitates further investigation, detectives ought to come up with strategic plans for how they anticipate to successfully collect information that will result in the identification and detention of the perpetrator. CIs employ a myriad of techniques to carry out follow-up inquiries, including motivational interviewing of victims, witnesses, and offenders, conducting concealed surveillance of a suspect, developing and managing informants, and using technological applications to simulate the crime scene events [23]. In the suspect management phase, e.g. the Police and Criminal Evidence Act of 1984 (an Act of Parliament which instituted a legislative framework for the powers of police officers in England and

Wales to combat crime and provided codes of practice for the exercise of those powers) should be observed in the identification of the probable crime perpetrator [25]. As per the Act, when the identity of the suspect is unknown, an available eyewitness should describe the individual before they are partaking in video and group identification. The eyewitness accounts, physical evidence, and any gathered information ought to engender some reasonable, objective bases to confirm that the isolated suspect has committed the offense.

2.2 Convolution Neural Network (CNN) and Text Classification

CNN is a type of artificial neural network which trains and tests multi-layers, as well as estimates weight sharing to decrease the network parameters [9]. It includes optional convolutional and pooling layers, whereby both layers collaborate to form several convolution groups. This allows extracting features layer by layer and enables the classification through various fully connected layers. A typical architecture of CNN is shown in Fig. 2.

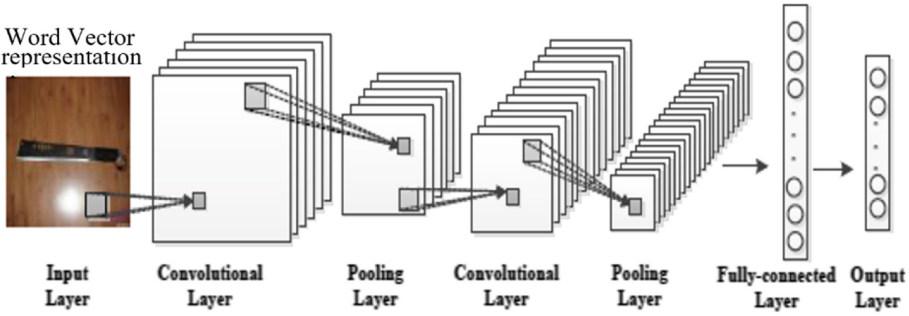


Fig. 2. Standard architecture of CNN [8].

The convolutional layer executes a convolution procedure, which includes a linear filter (i.e., kernel) that makes inner product operations at every input feature across the sliding window. The outcome of the inner product is employed by a non-linear activation function that generates an activation value related to the position of the input feature [8]. The pooling layer, called sub-sampling or down-sampling, reduces the number of parameters when the data matrix is too large, along with preserving useful information. The fully connected layer is used to encode a feature map matrix a vector space. The properties of CNN estimate the relation between the entire layers, layer connections and weight sharing [10]. This makes the CNN widely utilised in the domains of images classifications and objects detections, which is a motivation of using the CNN to classify textual crime types and their criminal acts.

Text classification is a well-known problem, which processes and classifies unstructured texts using Natural language processing (NLP). It converts unstructured data into structured data that can be handled by computer directly. Extracting the most significant features from textual data is essential to train and validate the CNN [9]. Several studies have used CNN for text classifications, for example, Conneau et al. [11] proposed a CNN

model for text classification, but it takes high processing time. In [12], the authors suggested a CNN model with and k-max pooling for text classification. The author applied a simple CNN algorithm for classifying short texts. The results revealed that the CNN algorithm achieves no less than conventional machine learning and NLP mechanisms in the semantic sentence classification [13].

Recent studies show that many technologies use deep learning CNN. For instance, Simonyan and Zisserman [26] reported that the convolutional networks have successfully facilitated extensive image and video identification, and it has formed the basis of ImageNet, a large public image archive, plus high-performance computing frameworks, like the graphical processing unit (GPU). Szegedy et al. [27] proposed VGG-Net, a deep CNN architecture with the hallmark of enhanced utilization of the computing resources within the network. Generally, the aforementioned convoluted networks are mainly for recognizing and tracking objects with class-specific bounding boxes. Saikia et al. [28] argue that crime scene photos and videos play an essential role in providing a visual report that allows CSIs to recreate the scene for further analysis and the identification of additional objects which may have been missed during the initial investigation. Nonetheless, owing to the presence of a massive volume of data, the activity of isolating trace evidence is exceedingly taxing for police departments. As such, the authors designed a faster Region-based CNN (R-CNN), a system that is intended to act analysing information collected at the crime scene via the object detection process. R-CNN was devised with the hypothesis that it is probable to extract intelligence by distinguishing objects found at the crime scene to assist the CSI, for instance, to associate diverse locations where criminal offenses were committed.

Lai et al. [14] developed a recurrent CNN technique that combines CNN and Recurrent Neural Network (RNN). The proposed technique applied RNN to extract context information and employed CNN to establish a semantic representation of text. Similarly, in [15], the authors proposed an RNN and CNN technique to identify continuous short texts, which revealed that CNN operates better than RNN. However, the algorithms above applied to short text classifications and consume high computational resources. This study uses CNN to classify long-textual crime types, such as larceny and credit stealing, and identify crime acts in short-time processing. This assists police in automating the process of crime investigation that contains a broad range of texts collected from victims, witnesses and questions of the police.

3 Proposed Cognitive Computing Technique-Based CNN for Crime Investigation

This section explains the proposed Cognitive Computing technique using CNN (CC-CNN) for classifying texts of crimes and identifying their acts. The proposed CC-CCN technique demands cleaning texts of crimes (i.e., tokenization) by removing special characters (e.g., !, ?, \) and converting upper letters to lower ones. Then, each sentence is padded to a specific sentence length (L), where padding sentences to a similar length is essential as it enables batching data with the same length while applying the CC-CNN technique.

Since texts of crime data have various lengths, a sliding window function is applied to each sentence to be converted to vectors (V_1, V_2, .., V_n), where n is the number of words in each class (C_1, C_2, …, C_d), where d is the number of crime classes such as robbery and assault. Suppose V_i has the k-dimensional word vector corresponding to the i^{th} word in the text of a class C_d. Short texts with a length less than n are filled so that all short texts have the same length. The vectorisation of words has been formulated by concatenating words of each record as follows:

$$S = V_1 \bigwedge V_2 \ldots . \bigwedge V_n \tag{1}$$

such that S is the sentences of each class, $\bigwedge$ is used for concatenating the words. The word vectors are merged as a matrix, which is used as input of the CNN model for extracting important features. A vocabulary index is developed to map every word in the sentences of a class to a specific integer value so that each sentence becomes a vector of integers.

The architecture of the proposed CNN techniques is represented in Fig. 3. The first layer converts words into low-dimensional vectors that include (n x k) representation of the sentence. The second layer executes convolutions over the embedded word vectors using many filter sizes, such as sliding over 2, 5 or 7 words at a time. A convolution operation includes a kernel function $T \in R^{k.h}$, where h is the height of the convolution sliding window, a window of h vectors can be converted and generate a new feature set that can be formulated by

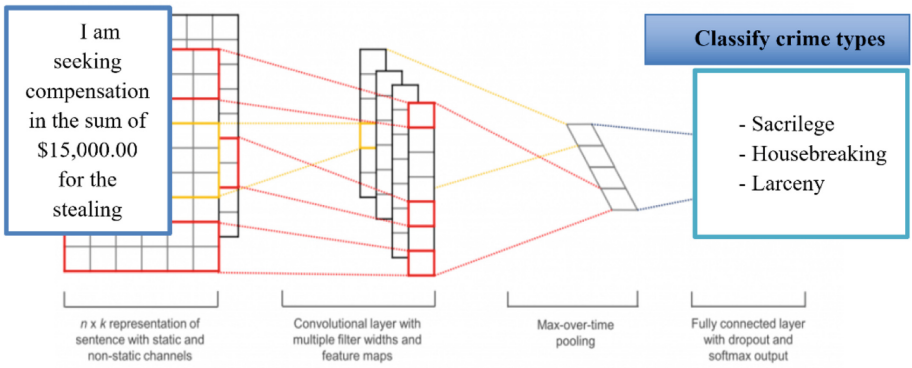


Fig. 3. Proposed CC-CNN technique for classifying texts of crime types.

$$C_i = f(T.V_{i:i+h-1} + b) \tag{2}$$

Where $b \in R$ is a bias value, f is a function, such as Sigmoid and Relu. The Relu activation function is used in this study as it is a linear function that decreases the parameters' dependency and solves the problem of overfitting.

After mapping the feature, the parameters are reduced by the max-pooling layer for producing the optimal features the outcome of the convolutional layer is max pooled

into a feature vector. Finally, all the produced local optimal features are linked through a fully connected layer whose output is the feature vector of each crime class (c). The fully connected layer uses dropout regularization to improve the performance and then classifies the outputs of crime types using a softmax function.

4 Experimental Results and Discussions

4.1 Datasets and Evaluation Setting

The proposed CC-CCN technique is evaluated using two datasets, namely sentiment polarity [16] and handcrafted text-crime datasets to provide a fair comparison. The sentiment polarity datasets involve negative and positive subsets of about 5000 movie-review records. The text-crime dataset includes more than 50000 records of ten crime types such as Sacrilege and Housebreaking. An example of a crime text is "I contacted City Police Station and about 7.10 pm that same day, Detective Wilson and Constable Smith arrived. I had a conversation with them and went with them whilst they examined my home.", which belongs to the crime class of "Sacrilege and Housebreaking".

The CC-CNN technique is trained and testing by dividing the datasets into 80% for the training set and 20% for the test set, with random states to ensure shuffling the data and avoid the overfitting problem. The evaluation criteria of accuracy, error rate and processing time are estimated to measure the performance of the technique.

4.2 Experimental Design

It is important to extract feature vectors from the text data to build the proposed CC-CNN technique. The text datasets are mapped into word vectors to be used as the input of the technique. Stochastic initialization and pre-trained approaches are used for pre-processing the input data. The stochastic initialization is used to input the datasets into the proposed technique for tokenizing and quantization texts. In the pre-trained approach, the text datasets are converted into word vectors using the word2vec embedding function [17]. This function produces word vectors by executing unsupervised learning on a broad range of text data.

The CC-CNN model uses four layers: an embedding layer, a convolution layer, max-pooling layer and softmax layer, based on the TensorFlow in Python [18]. The embedding Layer maps vocabulary word indices into low-dimensional vector representations. The TensorFlow's conv2d model produces a 4-dimensional tensor with 148 dimensions. The model includes 4 convolutions followed by max-pooling where filters of different sizes (i.e., 2, 5, 7) are applied. This is because every convolution generates tensors of various shapes, so that the model iterates through them, produce a layer for each of them, and then merges the outcomes into a feature vector.

A dropout layer is used to prevent neurons from co-adapting and forces them to learn individually important features. Using the feature vector from max-pooling, along with applying the dropout, predictions are produced as a matrix multiplication to select the class with the highest score and lowest loss. Then, a softmax function is applied to convert raw scores into normalized probabilities, which reflect the crime classes. The hyperparameters of the model are adjusted as listed in Table 1.

Table 1. Hyperparameters and their values of the CC-CNN model.

Hyperparameters	Value
Number of epochs	1
Batch size	39
Number of filters	32
Filter sizes	2, 5, 7
Embedding dimensions	60
Steps	200
Learning rate	10^-3
Dropout rate	0.45

4.3 Results and Discussions

The CC-CNN technique is validated using the sentiment polarity and crime-text datasets. The technique is compared with three techniques: Support Vector Machine (SVM) using Radial Basis Function (RBF), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM), in terms of accuracy, error rate and processing time, as demonstrated in Tables 2 and 3.

Table 2. Evaluation results using the sentiment polarity dataset.

Model	Accuracy (%)	Error (%)	Time (sec)
SVM (RBF)	85.30	14.7	190
RNN	87.82	12.18	188
LSTM	88.76	11.24	180
CC-CNN	91.86	8.14	173

Table 3. Evaluation results using the crime-text dataset

Model	Accuracy (%)	Error (%)	Time (sec)
SVM (RBF)	68.10	31.90	225
RNN	70.65	29.35	209
LSTM	73.38	26.62	215
CC-CNN	76.59	23.41	210

The CC-CNN technique achieves the highest accuracy and lowest error rates on both datasets. Using the results of the sentiment polarity dataset listed in Table 2, the technique achieves a 91.86% accuracy, an 8.14% error rate with roundly 173 s for training 5000 positive and negative classes. Following that, the LSTM and RNN techniques accomplish reasonable outcomes compared to the CC-CNN, while the SVM achieves the lowest outputs.

The CC-CNN technique reaches sensible outcomes using the crime-text dataset, as illustrated in Table 3. The technique attains a 76.59% accuracy, a 23.41% error rate with roundly 210 s for training about 50000 records with 10 crime classes. The LSTM and RNN mechanisms obtain sensible outputs in terms of accuracy, error rate and processing time compared with the CC-CNN technique, whilst the SVM achieves the lowest outputs which are the same behaviours of output using the sentiment polarity dataset.

There are several why the CC-CNN technique achieves better performances on the sentiment polarity dataset in comparison with the crime-text dataset. First, the crime text dataset includes 10 crime classes such as sacrilege, housebreaking, and larceny while the sentiment polarity dataset includes only two classes of the positive and negative behaviors of movie watching. Second, the number of records in the crime-text dataset is greater than the sentiment polarity dataset by 10 times with unbalancing between records of classes.

The results reveal that the proposed technique can sensibly classify crime types and identify acts of those crimes using text data. This technique would improve the manual process on crime investigation in police by learning text data of victims, witnesses and police questions to identify crime types. This enhances the processing time and helps the police officers and investigators to provide an accurate report that they use as evidence in a court of law.

5 Conclusion

This paper has proposed the use of the Cognitive Computing technique-based Convolution Neural Networks (CC-CNN) technique to classify crime types using text datasets. The proposed technique filters and analyses text data to identify classes from learning word vectors. The technique is trained and validated using two datasets of the sentiment polarity and handcrafted crime-text dataset. The experimental results revealed the proposed technique produces a reasonable performance on both datasets in terms of accuracy, error rate and computational time of training the model. From the results, it was revealed that by increasing the number of records of each class data, the model improves its performance compared with other machine learning models. The proposed technique could help the police to investigate crime types and discover the corresponding acts of crimes. This will improve the process of crime investigation that depends on specialist investigators who take too much time and cost. In the future, this work will be extended to apply more datasets of crimes and determine its validity in police systems. New data types, such as videos and images, will be used to investigate crime types.

Acknowledgments. We Acknowledge the AI-enabled Processes (AIP[1]) Research Centre and Spitfire Memorial Defence Grant (PS39150) for funding this research.

References

1. Higginson, A., Eggins, E., Mazerolle, L.: Police techniques for investigating serious violent crime: a systematic review. Trends Issues Crime Crim. Justice **539**, 1–13 (2017)
2. Loughnan, A.: The legislation we had to have?: the crimes (criminal organisations control) act 2009 (NSW). Curr. Issues Crim. Justice **20**(3), 457–465 (2009)
3. Police Investigation process, October 2019. https://www.victimsofcrime.vic.gov.au/police-investigation/the-investigation
4. Connor, M.A.: Professionalism in forensic archaeology: transitioning from "Cowboy of Science" to "Officer of the Court". In: Forensic Archaeology, pp. 33–42. Springer, Cham (2019)
5. Cronin, J.M., Murphy, G.R., Spahr, L.L., Toliver, J.I., Weger, R.E.: Promoting Effective Homicide Investigation (2007) (2019)
6. Stelfox, P.: Criminal investigation: an introduction to principles and practice. Willan (2013)
7. Scalia, V.: Martin O'Neill-Key (2018). Challenges in Criminal Investigation. Policing: J. Policy Pract. (2018)
8. Liu, Y., et al.: Crime scene investigation image retrieval with fusion CNN features based on transfer learning. In: Proceedings of the 3rd International Conference on Multimedia and Image Processing. ACM (2018)
9. Hu, Y., et al.: Short text classification with a convolutional neural networks based method. In: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE (2018)
10. Tzelepi, M., Tefas, A.: Deep convolutional learning for content based image retrieval. Neurocomputing **275**, 2467–2478 (2018)
11. Conneau, A., Schwenk, H., Barraul, L., et al.: Very deep convolutional networks for text classification. Assoc. Comput. Linguist. **1**, 107–1116 (2017)
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.A.: Convolutional neural network for modelling sentences. Assoc. Comput. Linguist. **1**, 655–665 (2014)
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
14. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2267–2273 (2015)
15. Lee, J.Y., Dernoncourt, F.: Sequential short-text classification with recurrent and convolutional neural networks. In: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 515–520 (2016)
16. Sentiment Polarity datasets, October 2019. http://www.cs.cornell.edu/people/pabo/movie-review-data/
17. Rong, X.: word2vec parameter learning explained. arXiv preprint arXiv:1411.2738 (2014)
18. CNN text classification TensorFlow, October 2019. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/
19. Fleming, J.: The pursuit of professionalism: lessons from Australasia. The future of policing, pp. 385–398. Routledge (2013)

[1] https://aip-research-center.github.io/.

20. Koper, C.S., Lum, C., Willis, J.J.: Optimizing the use of technology in policing: results and implications from a multi-site study of the social, organizational, and behavioural aspects of implementing police technologies. Policing: J. Policy Pract. **8**(2), 212–221 (2014)
21. Gehl, R., Plecas, D.: Introduction to criminal investigation: processes, practices and thinking. Justice Institute of British Columbia (2017)
22. Baber, C., Smith, P., Cross, J., Hunter, J.E., McMaster, R.: Crime scene investigation as distributed cognition. Pragmat. Cogn. **14**(2), 357–385 (2006)
23. Braga, A.A.: Moving the work of criminal investigators towards crime control. Harvard Kennedy School Program in Criminal Justice Policy and Management (2011)
24. Ariel, B., et al.: Report: increases in police use of force in the presence of body-worn cameras are driven by officer discretion: a protocol-based subgroup analysis of ten randomized experiments. J. Exp. Criminol. **12**(3), 453–463 (2016). https://doi.org/10.1007/s11292-016-9261-3
25. Parris, H.: The home office and the provincial police in England and Wales—1856–1870. In: The New Police in the Nineteenth Century, pp. 117–142. Routledge (2017)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
28. Saikia, M., Baruah, B.: Chaotic map based image encryption in Spatial domain: a brief survey. In: Proceedings of the First International Conference on Intelligent Computing and Communication. Springer, Singapore (2017)
29. Schiliro, F., Beheshti, A., Ghodratnama, S., Amouzgar, F., Benatallah, B., Yang, J., Sheng, Q.Z., Casati, F., Motahari-Nezhad, H.R.: iCOP: IoT-enabled policing processes. In: International Conference on Service-Oriented Computing, pp. 447–452. Springer, Cham, November 2018
30. Moustafa, N., Creech, G., Sitnikova, E., Keshk, M.: Collaborative anomaly detection framework for handling big data of cloud computing. In: 2017 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE, November 2017
31. Moustafa, N., Creech, G., Slay, J.: Anomaly detection system using beta mixture models and outlier detection. In: Progress in Computing, Analytics and Networking, pp. 125–135. Springer, Singapore (2018)
32. Keshk, M., Moustafa, N., Sitnikova, E., Creech, G.: Privacy preservation intrusion detection technique for SCADA systems. In: 2017 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE, November 2017
33. Beheshti, A., Schiliro, F., Ghodratnama, S., Amouzgar, F., Benatallah, B., Yang, J., Motahari-Nezhad, H.R.: iProcess: enabling IoT platforms in data-driven knowledge-intensive processes. In: International Conference on Business Process Management, pp. 108–126. Springer, Cham, September 2018

Abstraction-Based Outlier Detection for Image Data

Kirill Yakovlev[1], Imad Eddine Ibrahim Bekkouch[1], Adil Mehmood Khan[1(✉)], and Asad Masood Khattak[2]

[1] Innopolis University, Innopolis, Russian Federation
{k.yakovlev,i.bekkouch}@innopolis.university,
a.khan@innopolis.ru
[2] College of Technological Innovation, Zayed University,
Dubai, United Arab Emirates
asad.khattak@zu.ac.ae

Abstract. Data plays an important role in all stages of training, and usage of machine learning algorithms. Outliers are the samples in data that are generated by a "different mechanism" and belong to unexpected patterns that do not conform to normal behaviour. Outlier detection techniques try to deal with such undesirable events. There have been exceptional success of deep learning over classical methods in computer vision. In recent years a number of works employed the representation learning ability of deep autoencoders or Generative Adversarial Networks for outlier detection. Basically, methods are based on plugging representation techniques to outlier detection methods or directly reported employing reconstruction error as an outlier score. The error distributions of inliers and outliers may be still significantly overlapped. This could be associated with variation of samples inside the class, or cases with high outliers ratios, etc. In these cases, simply thresholding reconstruction errors may lead to misclassification. Although the produced representation is perhaps effective in representing the common features of the normal data, it is not necessarily effective in distinguishing outliers from inliers. We present a method that is based on constructing new features using convolutional variational autoencoder (VAE) and generate abstraction based on these features. To identify anomaly detection we tested two scenarios: utilizing VAE itself as well as using abstractions to train an additional architecture. Results are presented in the form of AUC-ROC using four benchmark datasets.

Keywords: Outlier detection · Convolutions · Variational autoencoder.

1 Introduction

Machine learning (ML) has become an important tool for solving problems in an overwhelming number of areas. Algorithms have been invented as effective tool

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 540–552, 2021.
https://doi.org/10.1007/978-3-030-55180-3_40

for certain types of learning tasks, and a theoretical understanding of learning is beginning to emerge. Many practical computer programs have been developed to exhibit useful types of learning, and significant commercial applications have begun to appear [20].

The application process of ML algorithms is usually associated with some consequent steps like model's conceptualization, data collection, model's training, assessment etc. Basically, all stages are directly affected by the data quality. In real world cases data are usually not as good as we would like to them be and could suffer from undesirable events (such as noise and errors) that may affect data interpretation, data processing, models that are based on these data as well as decisions made using these data.

However, sometimes the unusual data also can be due to rare, but correct, behavior, which often result in interesting findings, motivating further investigation. For these reasons it is necessary to develop techniques that allow us to identify such unusual events. Presumably we assume that such events may lead to generating samples by a "different mechanism" which indicates that these samples belong to unexpected patterns that do not conform to the normal behavior. In statistics and machine learning areas we used to call such objects as outliers [7].

In finance and banking areas, fraud detection is one such problem that is often formulated as an outlier detection problem. In order to protect their customers, organizations pay special attention to card usages that are rather different from typical cases. For instance, if a purchase amount is much bigger than common for a card owner or the transaction is initiated somewhere far from the owner's basic location, then it is considered as a suspicious activity. Obviously organizations want to detect such operations as soon as possible and contact the owner to validate a transaction. From the perspective of machine learning, we could consider such operation as abnormal which presumably has a different transaction pattern and is considered as an outlier case [10].

There have been attempts to develop outlier detection systems that could be effective for different tasks. Usually they utilize some properties of data and are focused on some specific anomalies. Probability-based techniques [8,16,19,23,27,31] utilize statistical models for outlining data model to distinguish between normal and abnormal events. Domain-based methods [2,13,26] try to define boundaries that separate outliers from a normal behaviour. Reconstruction approaches [1,11,14,28,30] usually apply predefined architectures and their ability to reconstruct a specific domain, where bad reconstruction is considered as an anomaly event. Information-theoretic [12,29] methods make a decision about an anomaly by utilizing concepts from an information theory.

Recently deep learning architectures have showed exceptional success over classical methods in computer vision tasks [3]. In the context of outlier detection, these methods usually utilize representation learning ability of deep autoencoders and GANs [9]. Outliers could be found using those representations as input for classical methods or directly employing a reconstruction error. Still there could the case of great variation in samples inside the class of images [5] or incidents when we are dealing with high outlier ratios [30]. Potentially this leads

to significant overlapping of error distributions between inliers and outliers. In these cases, simply establishing threshold for reconstruction error might lead to a misclassification problem. In this case, we can try to construct new samples based on extracted and generalized features. This information could be used to generate new synthetic samples from original data.

Sometimes in real-world problems we are dealing with data that do not contain all the possible examples of abnormal behaviour. It could be cumbersome or even not possible to catch all the deviation of a target class. One way to deal with that is to try to extract some additional information from the data that is already presented inside the data. Apparently every domain is described by some set of features or semantic information that we could extract for further purposes of classification. As an example, consider the problem of having a learner read a large collection of text and then solve object recognition problems. It may be possible to recognize a specific object class even without having seen an image of that object, if the text describes the object well enough. For example, having read that a cat has four legs and pointy ears, the learner might be able to guess that an image is a cat, without having seen a cat before. We call such type of learning as a zero-shot learning [9]. Given some semantic descriptions of object class, zero-short learning tries to accurately recognize objects of the unseen classes, for which no examples are available at the training stage, by associating them to the seen classes, from which labeled examples are provided [4]. Presumably it is possible to use this method in the context of outlier detection by trying to extract vital features from inliers and use it in some way to distinguish anomalies from normally behaved samples accordingly.

The main purpose of this research is to evaluate the use of reconstructed abstractions as a source of anomaly detection. For this purpose we used the principle of zero-shot learning. That is, we do not assume the availability of any outliers at training time. Our proposal is that it is possible to find a set of features from only inliers using a common feature extraction technique like convolution, then generalize those features and generate some new synthetic samples that is possible to use for a further outlier detection process.

This study is organized as the follows. The second section goes through some existed techniques for outlier detection and examples of applying abstractions for identifying anomalies. The third section explains abstraction concept as well as proposed method. The fourth section shows results of the study and the fifth chapter provides discussion aspect of the method and problem itself.

2 Related Work

At the moment outlier or anomaly detection is perceived as a separate area in the machine learning area. Anomaly detection is focused on detecting objects in data that does not fit well with the rest of the data. This problem covers different areas and applications such as fraud detection, surveillance, diagnosis, data cleanup, predictive maintenance, etc. [18,21,31]. Such a wide coverage of the spheres led outlier detection problem to become a huge area of study that contains big set of methods.

Probabilistic-based methods try to identify anomalies by utilizing distribution properties of the data. The simplest example is a Grubbs test that iteratively identify an outlier using a t-distribution assumption [31]. More complex methods are focused on adapting Gaussian-Mixture Model (GMM) method to model underlying joint distribution of the data using Gaussian distribution assumption [8,16,19,23,27]. However, in most real world situations the underlying distribution of the data is not known, which imposes limitations on using parametric distribution for modelling data patterns.

The other way of identifying outliers is a domain-based paradigm. This group of approaches are based on creating boundaries on the structure of the training dataset. These methods are typically insensitive to the specific sampling as well as density of the target class, because they describe the target class boundary, or the domain, and not the class density. Class membership of unknown data is then determined by their location with respect to the boundary [21]. This group of methods is primarily represented by Support Vector Machines (SVM) and its different variations [2,13] including One-Class SVM paradigm that aims to find a decision boundary to separate an inlier class from potential outliers [26].

Reconstruction-based outlier detection group is based on the modelling underlying data and assumes usage of some distance metric, e.g. reconstruction error, on the presented data, which is used as an outlier score. Practically we define the distance between the sample and the reconstructed output of the system and by using some threshold it is possible to identify an anomaly. There are bunch of neural network-based and subspace-based methods that can be trained in this way [21]. The common idea is to train neural network architectures for replication purposes [11,30]. Afterwards a trained architecture could be used to predict the given data. If an input point is not reconstructed well, which is associated with a high reconstruction error, these points can be considered as outliers. In this sense it is used an average reconstruction error as a novelty score. Subspace-based set of methods is focused on the data transformation into lower-dimensional space to identify features that catch the most of variance in the data like Principal Component Analysis (PCA) [14] to use distance in the principal component space to identify outliers or some types of competitive learning like Kohonen Maps that utilize a neighborhood function to preserve the topological properties of the data space [1,28] and use a Euclidean distance from a sample to points (e.g. neurons) that explain some patterns in the data.

Information theoretic methods are based on computation of measures such as entropy, relative entropy for information content of a data. The hypothesis is that novelties should significantly alter the information content of the "normal" dataset. One way to do this is to calculate metrics using the whole dataset and then gradually remove some subset of points. Subset of points whose elimination caused the highest difference in the metrics is considered to have anomalous data [21]. Initially the researchers used entropy to identify outliers in categorical data [12], which gradually was improved by employing the differential holoentropy [29].

3 Method

3.1 Abstraction Concept

Usually some objects, events are extremely rich in details. Let us assume that we can present some set of objects $X \in \{x_1, x_2, x_3, \dots\}$ in the form of an abstraction. The question is how to define an abstraction? From the general outlook, abstraction might be referred to the capability of removing inessential details and to identify a common "essence" inside variability [15]. Specifically we can assume an abstraction as a simplified representation based on generalized set of descriptors that define our set of objects. This claims that we can learn this representation R from the data using appropriate extraction techniques $R = f(X|descriptors)$. It means we can train a model $f(X)$ that could extract features, generalize them and generate simplified representations using those features. For the purpose of outlier detection, we can use a trained model assuming that it will create low-quality abstractions for outliers and higher-quality for inliers accordingly. As a second option, we could use synthesized abstractions as labels to inliers class to train a new model (e.g. deep autoencoder). The outlier detection would stay the same assuming that inliers and outliers produce abstractions with different level of quality. To assess the quality we can some score metric, which is used as an outlier score.

Thus, our initial idea is to identify and create an abstraction (or set of abstractions) that could explain our domain (the inliers) as accurate as possible and then use them for an outlier detection. The question is how to find and create this abstraction? One way to deal with that is to try to extract some additional information from the data that is already presented inside the data and base our abstractions on that. Apparently every domain is described by some set of features or semantic information that we could extract for further purposes. It means that practically we can extract those features using some feature extraction techniques like convolution [17]. It is widely used in an image processing for feature detecting without cancelling the spatial associations between pixels. It searches for a certain feature with the help of corresponding operator in a much larger pixel set. Convolution is an efficient way of feature extraction, skilled in reducing data dimension and producing a less redundant data set, also called as a feature map. Each kernel works as a feature identifier, filtering out where the feature exists in the original image. Eventually it produces a map whose altitude reveals how these features are distributed. In this research we adapted and used a set of deep convolutional layers which are architecturally identical to Deep Convolutional Generative Adversarial Networks (DCGAN) [22]. It is stated that this architecture allows to learn a hierarchy of representations from object parts to scenes and demonstrated its applicability for general image representations. This architecture is based on three principles:

1. It uses convolutional nets without using spatial deterministic functions (e.g. max pooling), which allows samples to learn its own spatial downsampling
2. After every convolution layer, there is a batch normalization which stabilizes learning by normalizing the input to each unit to have zero mean and

unit variance. This helps deal with training problems that arise due to poor initialization and helps gradient flow in deeper models
3. The ReLu activation follows by the batch normalization helps to obtain a bounded activation which presumably allows the model to learn more quickly to saturate and cover the color space of the training distribution.

Still the main issue is that features usually deviate in different domains as well as inside the domain itself. Convolution allows to extract features inside the domain, but it does not contain generalization properties of extracted features that abstraction is conceptually based on. Intentionally we want an abstraction that would have properties that explain the domain in a most efficient way. Presumably we could construct those features and use them to generate an abstraction accordingly. We must notice that new features do not contain a new information, because they are built using existing features, which are removed from the description. Feature construction is one of the most widely used in Machine Learning or Pattern Recognition area [25]. The process of adding new functions, relations and descriptors is usually called as The Constructive Induction. Conceptually, constructive induction is based on the changing of the language of representation. In our case, constructive induction aims to create new descriptors to preserve crucial information of the task. Consequently, we need a technique that could extract original features, create new features based on the generalization as well as generate intentional abstractions. Practically what we want is to combine convolutional techniques that would form joint distribution of our domain using extracted features, generalize them and generate abstractions accordingly. For this purpose we could utilize Convolutional Variational Autoencoder.

3.2 Convolutional Variational Autoencoder

An autoencoder is a neural network which projects data to and from a lower dimensional representation. It consists of two neural network models: encoder and decoder. The neural network is trained such that the output is as close to the input as possible, the data having gone through an information bottleneck: the latent space [9]. Presumably a vector of a latent space preserves topological properties on the data. However, in a high-dimensional case, e.g. image data, this could be insufficient to preserve complex patterns.

Variational autoencoder (VAE) instead preserves latent feature space using a distribution rather than a vector. This could help to catch additional deviation inside the domain as well as helps to generate new samples from it. Usually we approximate this distribution by the Gaussian distribution with zero mean and unit variance that looks as $\mathcal{N} \sim (0, I)$ for multivariate case. The generation in this case becomes more smooth and consistent moving from one class to another containing transitional samples with similar characteristics. Specifically this property of VAE helps to construct features by generalizing some areas of the domain in some boundaries, which could contribute in creating representational abstractions of the domain.

So far our model consists of encoder with convolution layers to extract features and a latent space assumed as a standard normal distribution. In order to generate abstractions we could use decoder part of the autoencoder, which is represented as a set of convolution layers in our model.

3.3 Structural Similarity Index Metrics and Outlier Detection

Presumably the proposed architecture can simultaneously perform the following tasks:

1. It consistently extracts features from the domain
2. It preserves features and generalize them inside the latent feature space
3. It generates abstractions from the latent space.

The next important step is an outlier detection. This can be done by two different scenarios. It is possible to use the trained VAE itself by relying on the reconstruction properties of the model or we can use generated abstractions to train a new model, e.g. Deep Autoencoder, and use it as an anomaly detector.

For the first scenario we can generate some set of abstractions from VAE assuming that they contain constructed features obtained during the training process. To identify anomalies, we can compare reconstructed samples with a chosen abstraction using a metric that could base the metric on similarities between abstraction and reconstructed sample. For this purpose we could use Structural Similarity Index Metrics [24]. Made up of three terms, the index estimates the visual impact of shifts in image luminance, changes in contrast, as well as any other remaining errors, collectively identified as structural changes. It is stated that SSIM outperforms such previous techniques as Mean Squared Error (MSE) and related PSNR (peak signal-to-noise ratio) in measuring the quality of natural images across a wide variety of distortions [6]. Presumably outliers should produce abstractions with noticeable distortions that can be catched by SSIM. We also can use SSIM to find a best abstraction among generated candidates that might be used for a comparing process.

The second option is to use generated abstractions as a target class to train another architecture. For example, we can train deep autoencoder by digesting samples from the inlier class and use abstractions as labels during the training process. During testing, outliers should generate abstractions that are less similar (or more distorted) to the abstractions generated by inliers accordingly.

4 Results

Both scenarios were applied on 4 datasets to analyze the method. The results are presented in the form of Area Under Receiver Operating Characteristic (AUC-ROC curve), which is used as a performance measurement for classification problem at various thresholds settings, together with a scatter plot to visually perceive the predicted clusters of inliers and outliers accordingly.

4.1 Datasets

Labeled Faces in the Wild. Labeled Faces in the Wild or LFW is a database of face photographs designed for studying the problem of unconstrained face recognition. This dataset was created and maintained by researchers at the University of Massachusetts, Amherst (specific references are in Acknowledgments section). It contains 13,233 images of 5,749 people. 1,680 of the people pictured have two or more distinct photos in the dataset. For outlier detection it was used deep-funneled version, as practically it provides superior results for face verification algorithms compared to other image types. We could use this dataset for experiments with outlier detection problem by choosing one person from the dataset and consider this class as our inlier group while other faces as outliers. Apparently in order to achieve a robust representation of inliers, it is reasonable to use a person with most number of photos in the dataset which in our case is George W. Bush represented by 524 different photos. Each sample is a symmetric image of 250 pixels for both dimensions that also includes three filters.

MNIST. The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. For our research it was taken about 30 000 samples of different digits and separated into train and test sets with 19 999 samples 9999 samples accordingly. Each digit was normalized to fit into 28×28 pixel bounding box with grayscale channel. For the purpose of outlier detection, it was chosen zero as an inlier class and outliers all other digits for a further research.

SVHN. The Street View House Numbers (SVHN) is a real-world image dataset that is usually used for developing machine learning and object recognition algorithms. Although it shares some similarities with MNIST where the images are of small cropped digits, SVHN incorporates an order of magnitude more labelled data (over 600,000 digit images). It also comes from a significantly harder real world problem of recognising digits and numbers in natural scene images. The images lack any contrast normalisation, contain overlapping digits and distracting features which makes it a much more difficult problem compared to MNIST. The dataset consists of 73,257 digits for training and 26,032 digits for testing. For the purpose of outlier detection, primarily both parts of the dataset were combined to extract samples of inlier class, which is one in our case. Then due to significantly big size of the dataset, it was decided to use only 20% of combined data for the training and 5% as a test data.

CIFAR-10. The CIFAR-10 dataset consists of 60000 32×32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. For the experiment, only 40% of the train dataset was used to extract an assumed inlier class, which is a car class and accordingly separated for train and prediction.

4.2 Test Results

In order to present results, test sets were created containing equal number of inliers and outliers. For a better visualisation, samples were ordered in a way that first half of test dataset would represent the inliers class, whereas the second part would contain outliers that formed by randomly chosen samples from different classes accordingly. It should be noted that the outlier classes were not used as a part of the training set.

Initially the proposed method was tested on the MNIST dataset. According to the obtained results, the first scenario outperformed the scenario with an autoencoder by 25% (Table 1). Results on the scatter plot (Fig. 1) confirms this point as well by showing almost separable clusters of SSIM Score for the first scenario. Despite some overlapping of the SSIM score, using variational autoencoder itself could be an considered as an effective approach. Overall MNIST showed the most convincing results comparing with other cases. The possible reason is that MNIST consists of one channel handwritten digits which are based on the simple patterns that are not difficult to learn for convolution layers. Usually it provides better results comparing with other cases [3]. For other experiments, the scenario with training a deep autoencoder where abstraction is used as labels during a training process outperformed the first scenario for LFW and SVHN datasets accordingly, while CIFAR-10 showed almost a parity for both scenarios. We could notice that there are cases with ROC < 0.5. Usually ROC of 0.5 defines a "random guess" level and cases with a lower level typically indicate some irrelevance of the applied technique or that the patterns to be classified were systematically different for train and test sets. In our case it could be associated with insufficient pattern explanation of an inlier class by generated abstractions.

Table 1. AUC-ROC scores

Datasets, % of the data	CVAE with AE	CVAE only
MNIST, 100%	0.70	0.957
LFW, 100%	0.735	0.42
SVHN, 20%	0.565	0.46
CIFAR-10, 40%	0.56	0.57

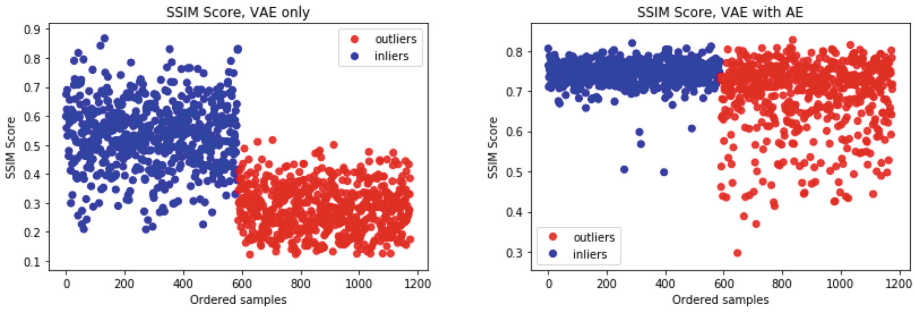


Fig. 1. MNIST dataset

5 Discussion

In this paper, we presented anomaly-based outlier detection based on zero-shot learning principle. This approach has some limitations as well as sides of possible improvements.

5.1 Abstraction Decision

Given that an abstraction is considered as something unclear from the practical point view, it imposes some challenges about how to understand that the model generates proper abstractions. It means there should be a stopping criteria for a variational autoencoder, as initially this architecture is used to generate accurately reconstructed data samples. Practically there should be a criteria that could assess that model has learned enough to generate presumable abstraction set. In our case, we used a common practice when we stop training when the training loss is equal to validation loss as well as a visual perception. The latter one means that we observed the reconstruction set and stop the algorithm at the moment when model starts to generate visually similar samples for different sample from the inliers set, as shown in Fig. 2. This could be considered as the model is trained and generalized enough to start generating possible abstractions. Practically for LFW dataset it might look as the following:

Fig. 2. Comparing original samples (top) and abstractions (bottom)

Still assessing the quality of abstractions remains challenging.

5.2 Comparison Process

During the test session, we choose one abstraction among generated candidates which was used to compare it with abstractions formed by test samples. In order to select an abstraction for comparison, SSIM was applied for each candidate and compared with the whole set of candidates. The abstraction with a highest average score was used further in the training process. We compared candidates among each other instead of original data, as some candidates might be identical to some original samples that could distort the results. In addition, it is more likely that a candidate with a higher score involves other properties of candidates as well. Still the best way of choosing right abstractions as well as number of candidates is still unclear. It is worth saying that the choice might considerably affect the final result.

5.3 Hyperparameter Tuning

During this research we did not utilize any advanced methods of identifying most relevant architectures as well as hyperparameters given the context of abstraction and its perceived ambiguity. In this research, we used the following set of parameters, see Table 2 for all datasets and architectures, which were found through grid search.

Table 2. Parameters set

Parameter	Value
Optimizer	Adam
Learning rate	0.001
Size of the latent space	20
Training epochs for VAE	95
Training epochs for AE	20
Number of candidates	500

6 Conclusion and Future Work

In this research we proposed ab abstraction-based approach for anomaly detection based on zero-shot learning principle for an image data. We tested two scenarios: the first assumes using the trained variational autoencoder itself for detecting anomalies, whereas the second is based on using synthesized abstractions to train a separate model (e.g. a conventional autoencoder). This approach involves many sub-elements that could significantly improve results. The main perspective is to investigate possible architecture improvements, as the presented architecture is not claimed as the most beneficial. There have been made a big progress in deep learning that produced many deep architectures that potentially

could extract inliers' features better. Also it is not clear whether we should use a static abstraction during a training of the following model. The appropriate usage of generated abstractions is also a possible way of improvement.

Results were presented in the form of ROC for several datasets. Overall proposed approach showed encouraging results for MNIST dataset, while controversial ones for other cases.

Acknowledgment. This research work was supported by Zayed University RIF Research Fund R19096.

References

1. Albertini, M.K., de Mello, R.F.: A self-organizing neural network for detecting novelties. In: SAC (2007)
2. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT 1992, New York, NY, USA, pp. 144–152. ACM (1992)
3. Cao, L., Yan, Y., Madden, S., Rundensteiner, E.: Outlier detection from image data (2019)
4. Changpinyo, S., Chao, W.-L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
5. Ciresan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
6. Dosselmann, R., Yang, X.D.: A comprehensive assessment of the structural similarity index. SIViP **5**(1), 81–91 (2011)
7. Escalante, H.J.: A comparison of outlier detection algorithms for machine learning. Program. Comput. Softw. **01** (2005)
8. Garcia, V., Nielsen, F., Nock, R.: Hierarchical gaussian mixture model
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org
10. Han, J., Kamber, M., Pei, J.: 12 - outlier detection. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining. The Morgan Kaufmann Series in Data Management Systems, 3rd edn, pp. 543–584. Morgan Kaufmann, Boston (2012)
11. Hawkins, S., He, H., Williams, G., Baxter, R.: Outlier detection using replicator neural networks **2454**, 113–123 (2002)
12. He, Z., Deng, S., Xu, X.: An optimization model for outlier detection in categorical data. In: International Conference on Intelligent Computing, pp. 400–409. Springer (2005)
13. Hu, W., Liao, Y., Rao Vemuri, V.: Robust anomaly detection using support vector machines (2003)
14. Jolliffe, I.: Principal Component Analysis. American Cancer Society (2005)
15. Kramer, J.: Is abstraction the key to computing? Commun. ACM **50**(4), 36–42 (2007)
16. Murali Krishna, N., Srinivas, Y., Lakshmi, P.V.: Truncated gaussian mixture model
17. Liu, Y.H.: Feature extraction and image recognition with convolutional neural networks. J. Phys. Conf. Seri. **1087**, 062032 (2018). IOP Publishing

18. Markou, M., Singh, S.: Novelty detection: a review-part 1: statistical approaches. Sig. Process. **83**(12), 2481–2497 (2003)
19. McNicholas, P.D.: Mixture model-based classification, October 2016
20. Mitchell, T.M.: Machine Learning, 1st edn. McGraw-Hill Inc., New York (1997)
21. Marco, A.F., Pimentel, D.A., Clifton, L.C., Tarassenko, L.: Review: a review of novelty detection. Signal Process. **99**, 215–249 (2014)
22. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
23. Reddy, A., Ordway-West, M., Lee, M., Dugan, M., Whitney, J., Kahana, R., Ford, B., Muedsam, J., Henslee, A., Rao, M.: Using gaussian mixture models to detect outliers in seasonal univariate network traffic. In: 2017 IEEE Security and Privacy Workshops (SPW), pp. 229–234, May 2017
24. Renieblas, G.P., Nogués, A.T., González, A.M., León, N.G., Del Castillo, E.G.: Structural similarity index family for image quality assessment in radiological images. J. Med. Imaging **4**(3), 035501 (2017)
25. Saitta, L., Zucker, J.-D.: Abstraction in artificial intelligence and complex systems, vol. 456. Springer (2013)
26. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
27. Scott, D.: Outlier detection and clustering by partial mixture modeling, January 2004
28. Tian, J., Azarian, M.H., Pecht, M.: Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm (2014)
29. Shu, W., Wang, S.: Information-theoretic outlier detection for large-scale categorical data. IEEE Trans. Knowl. Data Eng. **25**(3), 589–602 (2011)
30. Xia, Y., Cao, X., Wen, F., Hua, G., Sun, J.: Learning discriminative reconstructions for unsupervised outlier removal. In: The IEEE International Conference on Computer Vision (ICCV), pp. 1511–1519, December 2015
31. Zhang, Y., Meratnia, N., Havinga, P.: A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets. Praxis Der Informationsverarbeitung Und Kommunikation - PIK, January 2007

A Collaborative Intrusion Detection System Using Deep Blockchain Framework for Securing Cloud Networks

Osama Alkadi(✉), Nour Moustafa, and Benjamin Turnbull

School of Engineering and Information Technology, University of New South Wales at ADFA, Northcott Dr, Campbell ACT 2612, Canberra, Australia
o.alkadi@student.unsw.edu.au,
{nour.moustafa,benjamin.turnbull}@unsw.edu.au

Abstract. Security solutions, especially intrusion detection and blockchain, have been individually employed in the cloud for detecting cyber threats and preserving private data. Both solutions demand ensembled models-based learning that can alert the campaign of complex malicious events and concurrently accomplish data privacy. Such models would also provide additional security and privacy to the live migration of Virtual Machines (VMs) in the cloud. This would allow the secure transfer of one or more VMs between datacentres or cloud providers in real-time. This paper proposes a Deep Blockchain Framework (DBF) designed to offer security-based distributed intrusion detection and privacy-based blockchain with smart contracts in the cloud. The intrusion detection method is employed yet using a Bidirectional Long Short-Term Memory (BiLSTM) deep learning algorithm to deal with sequential network data and is assessed using the dataset of UNSW-NB15. The Privacy-based blockchain and smart contract methods are developed using the Ethereum library to provide privacy to the distributed intrusion detection engines. The DBF framework is compared with compelling privacy-preserving intrusion detection models, and the empirical results reveal that DBF outperforms the compelling models. The framework has the potential to be used as a decision support system that can assist users and cloud providers in securely and timely migrating their data in a fast and reliable manner.

Keywords: Intrusion detection · Privacy preservation · Blockchain · Deep learning · Cloud systems

1 Introduction

The lack of trust in the shared virtualised infrastructure deployed in cloud environments is a major impediment to achieving secure decentralised applications. Malicious cyber-attacks, such as Distributed Denial of Service (DDoS) and ransomware, target cloud-based platforms, exploiting the availability aspects of platforms. Such attacks are increasing in complexity and sophistication, resulting in disruptive consequences that can compromise data integrity, confidentiality and availability [1]. The ability to detect and

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 553–565, 2021.
https://doi.org/10.1007/978-3-030-55180-3_41

respond to such attacks is vital to conducting necessary mitigation and limiting any damage caused to cloud services. Intrusion Detection Systems (IDSs) are commonly deployed to monitor and discover those sophisticated attacks from endpoints of cloud networks, limiting the deployment of distributed IDSs that correlate security event alerts [2].

Cloud systems still face sophisticated attack scenarios that also increase with the emergence of blockchain [3]. For example, in June 2018, a number of blockchain crypto-currency such as, Bitcoin Gold, Zencash and MonaCoin have all fallen victim to a 51% attack, leading to loss of nearly 18 million worth of tokens [4, 5]. The malicious attackers were able to exploit each crypto-currency network and gain more than half of the total global mining hash rate. This vulnerability allowed them to double-spend transactions and affected the integrity of the whole network. Furthermore, in April 2016 an unknown attacker managed to drain more than 3.6 Million ether, the Ethereum currency, equating to $50 million USD from a decentralised autonomous corporation that is based on the blockchain and smart contracts, which rules the organisation [6]. In August 2016, another 120,000 units were stolen from the exchange platform Bitfinex in Hong Kong worth more than seventy million US dollars [7]. Furthermore, Bitfinex also suffered a temporary shutdown of service due to several security breaches, where it was targeted with DDoS and DSN attacks which a system with multiple virus-infected servers [8].

Applications-based blockchain has emerged in various domains to offer trust and data privacy services. Blockchain offers new opportunities by allowing participants to exchange transactions and share information while maintaining a degree of trust, integrity and enhanced transparency. Blockchain technology has numerous applications across different domains that go beyond the financial services and digital currency [9], including online voting [10], energy sector [11], Internet of Things (IoT) [12, 13], supply chain and manufacturing [14, 15], pharmaceutical and healthcare [16], big data [17], cyber security [18] and government services [19, 20] and many other disciplines. The history of data sharing is stored on immutable audit trails that can only be accessed by enterprises or privately hosted by cloud providers with specific permissions and trust criteria.

IDS and blockchain solutions are individually applied to the cloud systems to identify cyber-attacks and protect private data. IDSs in the cloud are basically classified based on deployment locations [21]. On the one hand, a Host-based IDS (HIDS) runs on a host system or VM to monitor and inspect audit data of operating systems such as memory and process audits. If the HIDS detects a malicious activity from an individual host or VM, the source IP is defined as access to the whole network to prevent user-to-root attacks from VM hopping and gaining access to another VM. On the other hand, a Network-based IDS (NIDS) system is placed at the infrastructure layer of cloud networks to monitor network traffic of all connected systems within a subnet. It can identify direct and indirect flooding, backdoor, port-scanning attacks, and suspicious malware activities [22].

Collaborative IDSs (CIDSs) are considered scalable and cost-effective to inspect various cloud nodes. One of the primary concerns in the cloud is the ability to maintain data protection and trust management between multi-cloud service providers [23]. The cloud system of being public, distributed and decentralised potentially leads to the challenge of trust as different components are controlled by different parties. Cloud providers are usually reluctant to share data or report intrusion events due to concerns about data confidentiality and privacy. It is quite difficult to measure the level of reputation among

untrusted participants. Another major challenge is insider attacks such as collusion and betrayal attacks, where malicious nodes collaborate to give false information and degrade the efficiency of alarm aggregation [23].

In addition to the discovery of attack events using CIDSs in the cloud, privacy-preserving techniques are widely used to transform, alter or conceal original data for protecting them from unauthorised access [24]. The blockchain and smart contract technologies are common types of privacy preservation that offer authentication and integrity to cloud elements. Blockchain technology addresses the lack of security trust and accountability through cryptography and consensus mechanisms [25]. Bitcoin is considered one of the first successful implementation of a distributed crypto-currency, where all transactions are processed without relying on third parties or agencies. This serves to safeguard data integrity and authenticity. The *s*ystem architecture of crypto-currencies is protected by extensive peer-reviewed cryptographic hash algorithms [26].

Ethereum is a cryptocurrency and a decentralised computing platform, that allows developers to program autonomous agents on a blockchain network in order to act as smart contracts [27]. However, a reliable model of trust for digital smart contracts implemented in a systematic approach in blockchains is currently lacking [18]. The smart contract should give users the flexibility and transparency to view the location of their migrated data within the blockchain ledger and the ability to track audit files between clouds. Recent reports have highlighted several security flaws and attacks in blockchain and its associated technology such as bitcoins and Ethereum [18, 28]. Developing a CIDS-based blockchain system in the cloud is essential to identify cyber-attacks and achieve data privacy of IDS engines deployed at various cloud nodes. This paper proposes a collaborative intrusion detection system based on a deep blockchain framework that achieves data security and privacy in cloud networks.

2 Related Studies

Blockchain solutions have been used in several studies to improve trust among collaborative IDSs in networks and cloud systems. For instance, Alexopoulos et al. [29] surveyed the methods of integrating CIDSs and blockchains. The authors introduced the concept of using blockchain techniques for enhancing the credibility of CIDSs. It is noted that characteristics of blockchain can benefit CIDSs in the ways of trusting each IDS and offering accountability and consensus methods. In [18], the authors also reviewed the significance of using blockchain and its theoretical approaches that would be employed to secure CIDSs. Liang et al. [28] proposed a decentralised and secured data provenance framework that offers tamper-proof data blocks. This framework allows data accountability and improves data privacy and prevents inference attacks from exploiting cloud systems. Wan et al. [30] proposed a hybrid consensus algorithm called Goshawk, which combines multiple layers of chain structure with many levels of PoW mining strategy and a ticket voting mechanism. This study presented that Goshawk is one of the early blockchain protocol with such properties having high efficiency, and robustness against 51% attacks.

Liu et al. [31] proposed a framework for IoT applications for securely sharing data collections. They suggested combining Ethereum blockchain with deep reinforcement

learning. Three main elements, environments, behaviour and incentives were used by the learning model. This increased the fairness ratio by more than 35% for the IoT software applications. In summary, the integration of blockchain and CIDS solutions would considerably improve security levels when they are deployed in cloud systems. Although IDS and privacy preservation have been widely used in the cloud. Integrating both systems could improve data security and privacy. In this paper, a deep blockchain framework is proposed to detect cyber-attacks using IDS-based on a BiLSTM model that can learn at any point in time from the surrounding context and can further protect private data using privacy preservation-based blockchain and smart contract.

3 Proposed Deep Blockchain Framework

3.1 Overall Systematic Architecture

A DBF is proposed to detect cyber-attacks and protect data in the cloud. The systematic architecture of the proposed framework includes four main components; cloud vendor, privacy-preservation based blockchain and smart contract, Central Coordinator Unit (CCU) and CIDS, as discussed below and illustrated in Fig. 1.

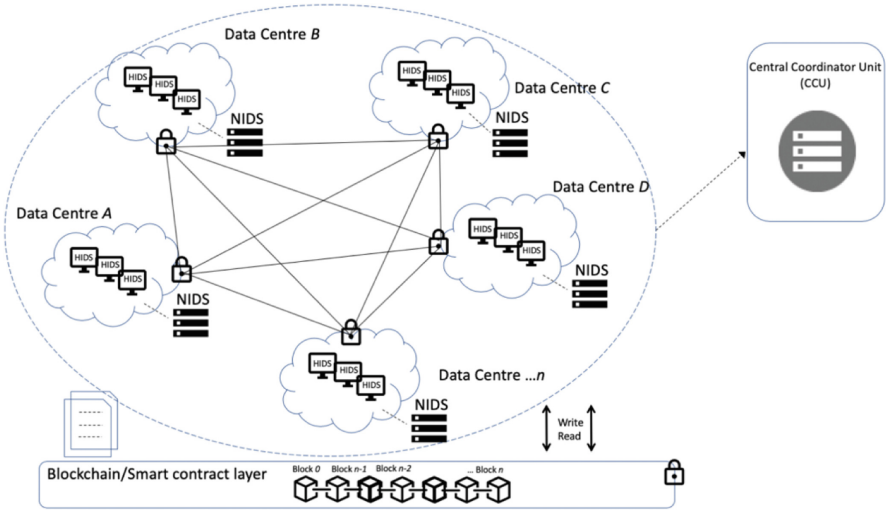


Fig. 1. Proposed cloud-based system architecture that includes the Deep Blockchain Framework (DBF). DBF would be deployed at NIDS and HIDS for cloud data centres. The blockchain/smart contract layer is designed to offer authentication and integrity to data and alerts generated by NIDS and HIDS.

- **Cloud Vendor -** different types of cloud vendors and data centres are represented. These are donated as data centres $A, B, C, \ldots, N$. They are identified as entities within a cloud network located in the blockchain network. These entities are expected to have enough cloud services to provide them with customer entities.

- **Privacy-preservation based Blockchain and smart contract -** this layer is different from a traditional cloud network as it incorporates a consortium blockchain; specifically, a distributed digital ledger containing the entire cloud transactions. This entity is replicated and stored in different nodes of the multi-cloud network, including CCU, datacentres or individual hosts. The proposed DBF has been constructed in a similar data structure to the Bitcoin's structure. Mining new blocks must be sufficiently rewarded during the process of adding a block to the blockchain.
- **Central Coordinator Unit (CCU) -** different cloud vendors may exchange data injected by malicious software activities, network traffic, and events logs among each other. The CCU acts as a Security Information and Event Management tool to store IDS audit logs and alerts. By leveraging the capabilities of the CCU, incoming data from different sources (i.e. cloud data centres) are analysed, filtered and correlated to distinguish between normal and abnormal events. This would enable network administrators to swiftly mitigate threats and increase security awareness for participants within the blockchain cloud network.
- **Collaborative IDS (CIDS) -** this entity orchestrates the verification of frames running on the cloud transaction network, and further ensures that they adhere to the specified rules. They consist of multiple IDSs deployed on large distributed networks or individual hosts that communicate with each other to detect coordinated cyberattacks and to prevent possible illegal actions. The primary purpose of CIDS is to enhance the overall detection accuracy of a single IDS node by correlating attack evidence over various sub-networks [1]. Thus, the CIDS enforces cooperation between different nodes would improve the capabilities to monitor sophisticated intrusions such as DoS, DDoS and malicious insiders [2].

3.2 Privacy-Preserving Using Blockchain and Smart Contract

Blockchain-based privacy preservation in the cloud extends the idea of a blockchain protocol, which operates on a peer-to-peer aspect to deliver encrypted data transactions or network nodes in a discrete way [32]. These encrypted messages form a chain of records or blocks that are stored on each participating cloud node confirming transaction integrity, so no records can be deleted or falsified from the ledger. Blockchain also enables the development of smart contracts, which are rule-based protocols that run on top of the blockchain network to enforce the negotiation of data usage policy (i.e., who can send IDS alerts and how they can be used) between involved cloud nodes. These policy-based rules define the raw data alerts produced by each IDS node as data transactions in the blockchain network. Collaborating IDS nodes can use the blockchain consensus mechanism to guarantee the validity and privacy of the stored alerts to create permanent and tamper-resistant data usage records.

Although blockchain and smart contract technologies remove the need for intermediaries for data protection, they are still inadequate to provide data privacy as all transactions are publicly accessible particularly in public network implementations [33]. To protect the confidentiality of smart contract data from disclosure by unauthorised users, we propose a hybrid method for privacy-preserving by integrating blockchain with a Trusted Executed Environment (TEEs). The TEE can be either hardware or software implementations that safeguard the confidentiality and integrity of applications [34].

Only permitted applications can read and write within the protected area of the CPU and memory. We propose an off-chain collaborative IDS that log alerts in the CCU, which is protected through the implementation of TEE. The CCU operates as an isolated environment running in parallel with the blockchain network as illustrated in Fig. 1 that can also withstand attacks from other higher privileged software such as the hypervisor. The goal of the proposed framework is to provide a confidential platform to execute smart contracts while still provide integration with existing cloud-based blockchain network such as Ethereum. To guarantee the confidentiality of the code and data of a smart contract, a secure remote assentation between the IDS nodes and CCU is established before transferring the contract.

4 Collaborative Intrusion Detection System Based Deep Learning

Our CIDS is built and deployed on could computing infrastructure due to its of their heterogeneous model and virtualised technology. Different cloud vendors may exchange event logs and shared alert data on malicious software activities amongst themselves. However, if such IDS systems are not trusted and appropriately integrated, the practical usage of shared data becomes limited. The unique characteristics of could computing present several challenges when designing a cloud-based CIDS. These desired characteristics include; efficient detection of insiders and outsiders' attacks while keeping false negatives (FN) and false positives (FP) at a minimum. The ability to scale dynamically across different data centre networks in the entire cloud. Furthermore, the framework would provide a maximum-security resistance to mitigate zero-day vulnerabilities and ensure data confidentiality, authentication, and integrity across all CIDS nodes [22].

4.1 Deep Learning Models for CIDS

This section presents the theories and fundamentals of the proposed DBF framework. RNNs are employed as IDS for detecting attacks from a blockchain-based cloud network [35]. It can be considered a powerful deep neural network that uses its internal memory within loops to deal with sequence data. The tackling of the temporal sequence is more relevant to the problem of intrusion detection, where the temporal patterns are present in user behaviour. This would improve anomaly or outlier detection which could be difficult to infer when relying only on the spatial domain (i.e. without accounting for time dimension). For this purpose, the internal state of the RNNs is used to process sequenced lists of crypto-records. The input sequence is handled in a series of time steps and associate memory is updated to produce a hidden state.

The standard RNN is defined as an artificial neural network with the capability of simulating discrete-time dynamical systems [36]. Such that, given a sequence of input vectors $x(t)$, a sequence of output vectors $y(t)$ are generated, where each time step $t \in \left[1, t_f\right]$ for a specific time interval t_f, one vector of the input sequence is processed, such that the internal state vectors are defined as $h_0(t) = x(t), \forall\ t \in \left[1, t_f\right]$ and $h_j(0) = x(t), \forall\ j \in [1, N]$. By applying the affine transformation a_j to the output vector of the previous layer and adding the linear transformation $V_j \in \mathrm{R}^{nj \times nj}$, the parameters of an RNN can be calculated using the cost functions below:

$$A_j(t) = W_j\, h_{j-1}(t) + V_j\, h_j(t-1) + b_j \tag{1}$$

$$h_j(t) = \sigma_j(A_j(t)) \tag{2}$$

where $h_j(t), A_j(t) \in R^{nj}$, and $z(t) = h_j(t)$, and W_j is the weight matrix from the input layer to the hidden layer, V_j is the weight matrix between two consecutive hidden states ($h_j(t-1)$ and $h_j(t)$), $_j$ is the bias vector of the hidden layer and σ_j is the activation function to generate the hidden state. The network output can be characterized as:

$$y(t) = \sigma_y(U_j h_j(\mathrm{t}) + b_\mathrm{y}) \tag{3}$$

where U_j is the weight matrix from the hidden layer to the output layer, $_\mathrm{y}$ is the bias vector of the output layer and σ_y is the activation function of the output layer. The parameters of the RNN is trained and updated iteratively via the back-propagation method. In each time step , the hidden layer will generate a value $y(t)$, and the last output $y(t_f)$ is the predicted network attacks.

4.2 Bi-directional Long Short-Term Memory (BiLSTM) Algorithm

One major drawback of RNNs is the inability of learning contextual information for an extended span of time caused by the vanishing gradient problem. This is mainly attributed to the prolonged temporal gap ranging from the time an input is obtained to making a decision. Weakening the ability of RNNs to learn from long distance dependencies [37]. Therefore, a LSTM algorithm, which is an extended version of RNNs-employed the idea of gates for related units [38]. It overcomes the vanishing gradient problem, and thus allows for preserving prolonged periods of contextual information.

In this context, the view of BiLSTM originated from bidirectional RNNs [39]. Bidirectional RNNs handles sequences of the input in forward as well as backward input directions by employing two different hidden layers. Figure 2 demonstrates a BiLSTM structure with multiple consecutive steps in time. BiLSTMs connects all hidden layers to the same output layer. A limitation of typical RNNs is that they can only use the previous context of the input data sequence. BiLSTMs compensates this by allowing for data to flow in both forward and backward directions [40].

The BiLSTM network estimates the forward hidden layer sequence output $\vec{h}(t)$, the output sequence of the backward hidden layer $\vec{h}(t)$ and the output layer $y(t)$ by reiterating the forward layer starting $t = 1$ to t_f, backward hidden layer since $\mathrm{t} = t_f$ to 1, and then updating the final value using following equations [41]:

$$\vec{h}(t) = H(W_{\vec{j}} X_t + V_{\vec{j}} h_{\vec{j}}(t-1) + b_{\vec{j}}) \tag{4}$$

$$\overleftarrow{h}(t) = H(W_{\overleftarrow{j}} X_t + V_{\overleftarrow{j}} h_{\overleftarrow{j}}(t-1) + b_{\overleftarrow{j}}) \tag{5}$$

$$y(t) = U_{\vec{j}} h_{\vec{j}}(t) + U_{\overleftarrow{j}} h_{\overleftarrow{j}}(t) + b_y \tag{6}$$

The final output vector, $\mathbf{y}(t)$ is calculated as

$$\mathbf{y}(t) = \sigma_y(\vec{h}, \overleftarrow{h}) \tag{7}$$

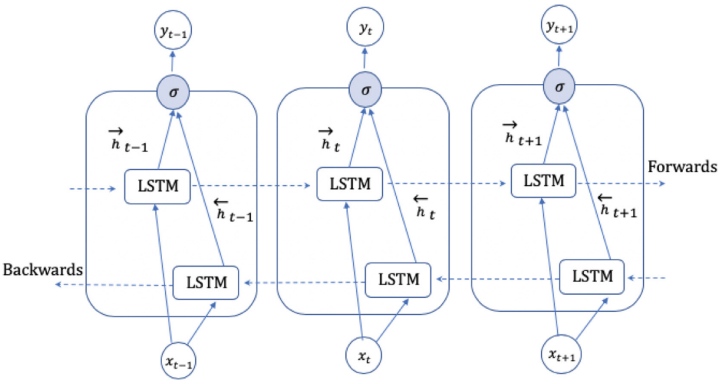


Fig. 2. Architecture of LSTM with three consecutive layers

The σ_y function concatenates the output sequences of the neurons in the hidden layers and cloud be one of four operations: add, multiply, average and concatenate. For the RNN training stage, the BiLSTM was employed as defined by Schuster et al. [39] and learning representations by back-propagating errors defined by Rumelhart et al. [42] (see Fig. 2).

5 Experimental Results and Evaluations

5.1 Experimental Design

To evaluate the efficiency of the proposed DBF, a private blockchain was created using Ethereum [27], which is an open-source blockchain platform where users can establish and deploy private blockchains within organisation datasets and pre-processing module used for evaluation. The Ethereum network provides a virtual machine runtime environment to run and execute the smart contracts. The Google Colab [43] cloud service was used for the experiments using TensorFlow library Keras for deep leaning on three types of hardware accelerators CPU, GPU and TPU for implementing the RNN-based BiLSTM model as CIDS. The CIDS model was compared with other machine learning methods; specifically Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB) and Mixture Localisation-based Outliers (MLO) [22]. The evaluation of the proposed DBF for intrusion detection was conducted using the network dataset of UNSW-NB15 [44]. The accuracy (AC), Detection Rate (DR) and processing time are used to evaluate the CIDS performance.

5.2 Results and Explanations

The RNN-based BiLSTM model as IDS was trained and validated using a large amount of labeled data in the training phase. This gives the model more information to be able to extract enough reliable features to act as a baseline for the training phase. The model was configured by an input layer fed from the UNSW-NB15 dataset, two hidden layers with hidden nodes = *60*, a *tanh* activation function, and the output layer includes a *softmax*

activation function to predict the two classes of normal and attack types. The model was adapted using the hyperparameters of *loss = 'binary_correntopy', optimizer = 'adam' batch_size = 100, epochs = 200, metrics = 'accuracy'*.

The performance evaluation of the IDS was conducted on the UNSW-NB15 dataset, with the overall accuracy, demonstrated for various Hidden Nodes (HN = {10, 20, 30, 40, 50, 60}), as listed in Table 1. The training and testing computational processing times using different hardware accelerators (CPU, GPU, TPU) are evaluated as well. The proposed framework with HN = 20 requires approximately computational processing time of 400, 69, 61 s of CPU, GPU and TPU training around 14,000 data observations for the UNSW-BN15 dataset and 22, 11, 11 s for testing on CPU, GPU and TPU. It is evident that the relation between estimated accuracy and number of hidden nodes is proportional; with the best accuracy (99.41%) achieved with 60% hidden nodes for the UNSW-BN15 dataset.

Table 1. Accuracy of BiLSTM RNN using UNSW-BN15 dataset with different number of hidden nodes and hardware accelerators

HN	Accuracy	Training time (s)			Testing time (s)		
		CPU	GPU	TPU	CPU	GPU	TPU
10	97.26%	200.3	59.2	54.3	13.3	8.1	7.2
20	97.61%	400.6	69.1	61.2	21.8	11.4	10.7
30	97.94%	891.2	89.5	84.7	34.3	17.4	16.9
40	98.21%	991.2	98.2	122.4	67.5	20.4	19.4
50	98.58%	1120.2	138.7	132.5	71.4	28.1	29.4
60	99.41%	1901.2	190.3	186.4	89.1	43.6	42.7

Central Processing Unit (CPU), Graphics Processing Unit (GPU) and Tensor Processing Unit (TPU)

The accuracy of the proposed model using the UNSW_NB15 and employing different hardware accelerators are shown in Table 1. The accuracy improves as the number of hidden nodes increases. The results revealed that with increasing hidden nodes size from 10 to 60 the accuracy of the model increases from 97.26% to 99.41% using the UNSW-NB15 dataset. The proposed model can detect different attack types from the dataset in an average of 95%–99% as demonstrated in Table 2. Various types of attacks, such as DDoS TCP, DDoS UDP, DDoS HTTP, DoS TCP, DoS UDP, and DoS HTTP, can be reliably distinguished with accuracy exceeding 99%. The proposed IDS system can also detect the remaining malicious activities that attempt to disrupt cloud services, such as reconnaissance, port scanning, keylogging, and data theft attacks, with reasonable detection rates.

Table 2. Detection rates of attack types from UNSW-BN15 dataset

Attack type	Dataset type
	UNSW-BN15
DoS http	99.79%
DoS udp	99.89%
DoS tcp	99.71%
Backdoor	97.49%
Exploits	97.85%
Analysis	95.65%
Reconnaissance	96.88%
Worms	94.75%
Shellcode	97.90%
Port scanning	95.98%
OS Fingerprinting	93.14%
DDoS http	99.89%
DDoS udp	99.95%
DDoS tcp	99.59%
Keylogging	92.85%
Data theft	98.54%

5.3 Discussions

The proposed framework benchmarks well against other approaches for preserving privacy and identifying attack behaviours, that can be attributed to the multilevel abstraction of the data in the IDS model design. The two-fold privacy model of blockchain and smart contracts can achieve perfect protection by validating data transactions and extracting features from the source data for training and validation of the IDS model. In the first phase of the privacy-based blockchain, data integrity is verified and records are checked for possible poisoning by the applied hash chain, making malicious alteration of records infeasible (i.e. highly expensive computationally). In the second phase, the data is encoded by validating unusual behaviour detection as an example of efficiency and performance measuring.

The RNN-based BiLSTM technique is selected due to its inherent characterisation of the Spatio-temporal context, and with its synchronous and limited memory properties, it permits for accurate future conditions predications with all past and last input scenarios. Results show that the RNN-based BiLSTM technique can efficiently classify legitimate and suspicious records after encoding the data using the two-phase privacy-preserving methods. The proposed system could be easily deployed by cloud computing centres. This could happen when the network data is called a distributed database management system to collect important features from different network nodes. It can be deployed

as a service as it has a low computational overhead. This advantage stems from the fact that its potential design is based on timely estimating features' parameters of the CIDS in cloud networks. The future extension of this work will include implementing the framework on various real-world datasets to evaluate its scalability and utility.

6 Conclusion and Future Work

This paper has introduced a collaborative intrusion detection system-based DBF for the identification of cyber-attacks. It is based around privacy preservation and designed specifically for the cloud. The DBF framework has two principle goals; privacy preservation and intrusion detection. The privacy-preserving method comprises a hybrid approach by firstly combining blockchain with a trusted executed environment to provide confidentiality of smart contracts, while simultaneously maintaining integrity and availability. Subsequently, the network data is encoded through a deep neural network model. The hybrid privacy-preserving method attains better performance compared with recent works, as it is resilient to inference and data poisoning attacks. The second method of intrusion detection is based on a BiLSTM algorithm that was evaluated on the UNSW-NB15 dataset for classifying attack events that exploit cloud networks. The results revealed that the proposed intrusion detection method can outperform other techniques in terms of accuracy and detection rate. The proposed framework enables exchanging data between cloud simpler, safer and more transparent while also significantly reducing overheads. It will further act as a decision support tool to help users and cloud providers securely migrate their data in a fast and flexible manner. The future extension of this work will include applying the framework on different real-world datasets to evaluate its scalability and utility.

References

1. Patel, A., Taghavi, M., Bakhtiyari, K., Júnior, J.C.: An intrusion detection and prevention system in cloud computing: a systematic review. J. Netw. Comput. Appl. **36**(1), 25–41 (2013)
2. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016)
3. Alkadi, O.S., Moustafa, N., Turnbull, B., Choo, K.-K.R.: An ontological graph identification method for improving localisation of IP prefix Hijacking in network systems. IEEE Trans. Inf. Forensics Secur. **15**, 1164–1174 (2019)
4. Aitzhan, N.Z., Svetinovic, D.: Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Depend. Secure Comput. **15**(5), 840–852 (2016)
5. Sayeed, S., Marco-Gisbert, H.: Assessing blockchain consensus and security mechanism against the 51% attack. Appl. Sci. **9**(9), 1788 (2019)
6. Fraga-Lamas, P., Fernández-Caramés, T.M.: A review on blockchain technologies for an advanced and cyber-resilient automotive industry. IEEE Access **7**, 17578–17598 (2019)
7. Liu, J., Liu, Z.: A survey on security verification of blockchain smart contracts. IEEE Access **7**, 77894–77904 (2019)
8. Baldwin, C.: Bitcoin worth $72 million stolen from Bitfinex exchange in Hong Kong. https://www.reuters.com/article/us-bitfinex-hacked-hongkong-idUSKCN10E0KP. Accessed July 2019

9. Peters, G.W., Panayi, E.: Understanding modern banking ledgers through blockchain technologies: future of transaction processing and smart contracts on the internet of money. In: Banking Beyond Banks and Money, pp. 239–278. Springer (2016)
10. Hardwick, F.S., Gioulis, A., Akram, R.N., Markantonakis, K.: E-voting with blockchain: an e-voting protocol with decentralisation and voter privacy. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1561–1567 (2018)
11. Knirsch, F., Unterweger, A., Eibl, G., Engel, D.: Privacy-preserving smart grid tariff decisions with blockchain-based smart contracts. In: Sustainable Cloud and Energy Services, pp. 85–116. Springer (2018)
12. Khan, M.A., Salah, K.: IoT security: review, blockchain solutions, and open challenges. Future Gener. Comput. Syst. **82**, 395–411 (2018)
13. Fernández-Caramés, T.M., Fraga-Lamas, P.: A review on the use of blockchain for the internet of things. IEEE Access **6**, 32979–33001 (2018)
14. Tian, F.: A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things. In: 2017 International Conference on Service Systems and Service Management, pp. 1–6 (2017)
15. Abeyratne, S.A., Monfared, R.P.: Blockchain ready manufacturing supply chain using distributed ledger. Int. J. Res. Eng. Technol. **5**(9), 1–10 (2016)
16. Yue, X., Wang, H., Jin, D., Li, M., Jiang, W.: Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. **40**(10), 218 (2016)
17. Karafiloski, E., Mishev, A.: Blockchain solutions for big data challenges: a literature review. In: IEEE EUROCON 2017-17th International Conference on Smart Technologies, pp. 763–768 (2017)
18. Meng, W., Tischhauser, E.W., Wang, Q., Wang, Y., Han, J.: When intrusion detection meets blockchain technology: a review. IEEE Access **6**, 10179–10188 (2018)
19. Ølnes, S., Ubacht, J., Janssen, M.: Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. Elsevier (2017)
20. Alketbi, A., Nasir, Q., Talib, M.A.: Blockchain for government services—use cases, security benefits and challenges. In: 2018 15th Learning and Technology Conference (L&T), pp. 112–119 (2018)
21. Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., Rajarajan, M.: A survey of intrusion detection techniques in cloud. J. Netw. Comput. Appl. **36**(1), 42–57 (2013)
22. AlKadi, O., Moustafa, N., Turnbull, B., Choo, K.-K.R.: Mixture localization-based outliers models for securing data migration in cloud centers. IEEE Access **7**, 114607–114618 (2019)
23. Li, W., Meng, W., Kwok, L.-F., Horace, H.: Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model. J. Netw. Comput. Appl. **77**, 135–145 (2017)
24. Bernabe, J.B., Canovas, J.L., Hernandez-Ramos, J.L., Moreno, R.T., Skarmeta, A.: Privacy-preserving solutions for Blockchain: review and challenges. IEEE Access **7**, 164908–164940 (2019)
25. Olleros, F.X., Zhegu, M.: Research Handbook on Digital Transformations. Edward Elgar Publishing, Cheltenham (2016)
26. Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. **18**(3), 2084–2123 (2016)
27. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, vol. 151, no. 2014, pp. 1–32 (2014)

28. Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., Njilla, L.: ProvChain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 468–477. IEEE Press (2017)
29. Alexopoulos, N., Vasilomanolakis, E., Ivánkó, N.R., Mühlhäuser, M.: Towards blockchain-based collaborative intrusion detection systems. In: International Conference on Critical Information Infrastructures Security, pp. 107–118. Springer (2017)
30. Wan, C., et al.: Goshawk: a novel efficient, robust and flexible blockchain protocol. In: International Conference on Information Security and Cryptology, pp. 49–69. Springer (2018)
31. Liu, C.H., Lin, Q., Wen, S.: Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning. IEEE Trans. Ind. Inf. **15**(6), 3516–3526 (2018)
32. Liang, G., Weller, S.R., Luo, F., Zhao, J., Dong, Z.Y.: Distributed blockchain-based data protection framework for modern power systems against cyber attacks. IEEE Trans. Smart Grid **10**(3), 3162–3173 (2018)
33. Huebsch, R., Chun, B., Hellerstein, J., Loo, B., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I., Yumerefendi, A.: The architecture of PIER: an Internet-scale query processor. In: Proceedings of 2nd Biennial Conference on Innovative Data System Research, pp. 28–43 (2005)
34. Dalbehera, P., Andersson, S., Varshney, R., Wallin, P.: Dynamic configuration of trusted executed environment resources. Google Patents (2016)
35. Mishra, P., Khurana, K., Gupta, S., Sharma, M.K.: VMAnalyzer: malware semantic analysis using integrated CNN and bi-directional LSTM for detecting VM-level attacks in cloud. In: 2019 Twelfth International Conference on Contemporary Computing (IC3), pp. 1–6. IEEE (2019)
36. Yadav, R.M.: Effective analysis of malware detection in cloud computing. Comput. Secur. **83**, 14–21 (2019)
37. Bengio, Y., Boulanger-Lewandowski, N., Pascanu, R.: Advances in optimizing recurrent networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8624–8628. IEEE (2013)
38. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
39. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
40. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
41. Cui, Z., Ke, R., Wang, Y.: Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143 (2018)
42. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
43. Google. Google Colaboratory. https://colab.research.google.com. Accessed May 2019
44. Moustafa, N., Hu, J., Slay, J.: A holistic review of network anomaly detection systems: a comprehensive survey. J. Netw. Comput. Appl. **128**, 33–55 (2019)

A Deep Learning Cognitive Architecture: Towards a Unified Theory of Cognition

Isabella Panella(✉), Luca Zanotti Fragonara, and Antonios Tsourdos

Cranfield University, Cranfield MK43 0AL, UK
i.p.panella@cranfield.co.uk

Abstract. This work suggests a novel approach to autonomous systems development linking autonomous technology to an integrated cognitive architecture with the aim of supporting a common artificial general intelligence (AGI) development. The paper provides a summary of strengths and weaknesses of some of the most known cognitive architecture and highlights how to support a generic artificial intelligent approach rather than ad hoc solutions. It also proposes objective evaluation criteria to assess a cognitive architecture. Finally, the proposed cognitive architecture is introduced: a Deep-Learning Artificial Neural Cognitive Architecture (D-LANCA), which aims to overcome current limits of cognitive frameworks for autonomous systems with the view to create a common artificial general intelligent (AGI) cognitive approach across industries.

Keywords: Cognitive architecture · Deep neural networks · Deep learning

1 Introduction

1.1 A Subsection Sample

The success of autonomous software platforms to be a ubiquitous enabling capability delivering significant value across a range of industries and the driving requirement to implement is dependent upon the abilities for autonomous technologies to reach a high level of cognition in mimicking the human mind. This implies the design and implementation of computational routines able to combine cognitive abilities in an integrated manner, what is referred to the creation of general intelligent systems [1, 2]. The first challenge it is presented by the definition of "intelligence" within an autonomous system. In order to design a system, the definition of what the system is needs to be clear and associated to mathematical and physical representations. The need for the system to be adaptable to its environment. Therefore, the author suggests the following design definition for an autonomous system. Intelligence in autonomous platforms can be defined as the ability for a system to adapt to its environment and survive. Artificial Intelligence (AI) can be considered as a technology field that aims to embed self-modulated responses into machines, which will enable them to be self-driven towards a primary goal: the system survival in itself. AI can be thought as the combination of various technologies that will provide machines with the sense of survival, hence what we described as adaptation capabilities. Adaptation requires the machines to:

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 566–582, 2021.
https://doi.org/10.1007/978-3-030-55180-3_42

- Sense the environment and themselves;
- Create an inner representation of what they sense;
- Be able to reason and make inferences about the environment;
- React to the environment;
- Learn and update knowledge;
- Re-plan their course of action;
- Actuate the new plan.

For each of the points above, it is possible to associate a mathematical representation to support the implementation of each functional requirement in the design of an intelligent system.

It is possible to distinguish two kind of "adaptability" within an autonomous system: *operational* and *behavioral*.

Operational adaptability enables platforms to perform procedural tasks autonomously. It refers to the "what" to perform, ranging from the sensor payload selected for a given task to the dynamic stability of the system.

Behavioral adaptability refers to the functionalities embedded within a system that will enable it to decide how to perform it and to adapt its behavior and reaction to the environment it operates in. This enables the system to perform autonomously in any environment condition, it determines the "how" a procedural task should be carried "how" for the given situation picture.

Current approaches to autonomous systems have seen the implementation of *ad hoc* and often divided approaches to this problem, with the implementation of specific solutions to the challenge of intelligent behavior, such as planning, computer vision, linguistic, problem solving, etc. This fragmented approach has led to significant advances in computer science but has diverted the attention from the so called Unified Cognition Theory envisaged by Newell [1–6, 10–15], the creation of an artificial general intelligence (AGI).

The paper is organized as following. In Sect. 2, an analysis of known cognitive architectures with their strengths and weaknesses is provided, with a summary of the gap analysis in current provided solutions. In Sect. 3, we present a suggestion on how to evaluate objectively cognitive architectures. Section 4 presents the cognitive architecture, a Deep-Learning Artificial Neural Cognitive Architecture (D-LANCA) that could address the challenges identified in Sect. 2. Finally, we present conclusions and recommendations.

2 Cognitive Architecture: Strengths and Weaknesses

Cognition implies the ability of interpreting the reality and understand how things could be in a future state to support the decision-making process. It also highlights the need to "remember" past events to support forecasting future events outcomes. Therefore, as reported by Vernon, D. et al. [3], cognition "allows the system to act effectively, to adapt, and to improve".

Cognition can be considered a process through which "the system achieves robust adaptive, anticipatory, autonomous behavior, entailing embodied perception and action." [3].

Cognitive architectures were proposed by Allen Newell, one of the founders of Artificial Intelligence (AI), in 1980 in an effort to avoid "theoretical myopia" in his search for a Unified Theory of Mind [4–6].

"Cognitive Architectures" (CA) in machines applications refer to abstract model of cognition in artificial agents with the associated software functionalities linked to their implementation in autonomous systems through Artificial Intelligence (AI) methodologies. As stated by Langley et al. [6], they can be considered "the underlying infrastructure for an intelligent system". Profanter, S. [7], defines a cognitive architecture as "a blueprint for an intelligent system", whereby "blueprint" refers to the design characteristics of the framework, such as the assumptions made to support representation, its memory characteristics, and the process used to operate those memories. Their main role is to realize artificial systems that exhibit intelligent behaviors in general setting mimicking human behavior [7]. The objective is to model complex and higher-level functionalities, such as language, reasoning, problem solving, decision making, or planning, which involve complex knowledge structures and complex information extraction methods [8].

Extensive surveys of cognitive architectures have been carried out by [6, 8, 13, 17], and [13] among others. A summary of the most adopted cognitive architectures' implementation has been presented by the authors in the table below (Table 1). Herein, the architectural approach has been highlighted as well as the learning algorithms. The rational for highlighting the learning algorithms is that they represent the ability of the system to behaviorally adapt to the environment. In the table, we also highlight the architecture characteristics and we then assess the weakness that each architecture presents with respect to its application to autonomous platforms.

2.1 Gap Analysis of Current Cognitive Architectures

It is possible to observe from the analysis in Table 1 that whilst significant achievements have been accomplished over time by the various cognitive architectures' developments, there are still open research questions and challenges related to:

- Integration of distinct cognitive functions;
- Computational efficiency of the platform;
- Parallel processing of data - even though current architectures have some commitment to parallelism, especially in memory retrieval, they tend to rely on one or few decision modules [6];
- Integration of heterogeneous knowledge representation;
- Integration of heterogenous reasoning and cognitive functions;
- Integration of planning, acting, monitoring and goal reasoning;
- Knowledge problem - knowledge acquisition, knowledge size, and homogeneity and homogeneous typology of encoded knowledge used by the agents in reasoning, decision making, learning, planning, etc. [20, 22]. Knowledge acquisition is the ability

Table 1. Cognitive architectures characteristics and weaknesses [8–23]

	Approach	Learning algorithms	Characteristics	Weaknesses
CLARION – Connectionist Learning with Adaptive Rule Induction On-line	Artificial General Intelligence (AGI) – software agent-based architecture Knowledge and experiences are represented in CLARION within an implicit-explicit dichotomy: using chunks and rules (for explicit knowledge), and neural networks (for implicit knowledge) It is based on a four-way division of implicit versus explicit knowledge and procedural versus declarative knowledge It addresses both top-down learning (from explicit to implicit knowledge) and bottom-up learning (from implicit to explicit knowledge) It is an hybrid paradigm architecture	Reinforcement Learning Hebbian Learning Bayesian update Gradient descent methods Learning of new representations through new chunks, new rules, and new NN representations Hypothesis testing rule learning for explicit learning Bottom-up rule learning – implicit to explicit learning	The working memory is a separate structure Semantic, episodic, and procedural memory are both implicit and explicit (chunks/rules, NN) It has a reward system in the form of a motivational subsystem and a meta-cognitive subsystem (MCS determines rewards based on the MS) Support autonomous learning from new representation	No general and cross-domain knowledge – difficulties in dealing with variety of data coming from different sources The dual-layered conceptual information does not provide the possibility of encoding (manually or automatically via learning cycles) the information in terms of the heterogeneous classes of representations Possible co-existence of different representations of prototypes, exemplars and theories (and the interaction among them) is not addressed
SOAR – State, Operator and Result	Artificial General Intelligence (AGI) – software agent-based architecture Rule based architecture designed to model general intelligence Knowledge and experiences are represented using rules (procedural knowledge), relational graph structure (semantic knowledge), and episodes of relational graph structures (episodic memory) It is a symbolic architecture	Reinforcement Learning Chunking to form new rules – production rules (procedural long-term memory) Bayesian update Complex rule sets in planning, problem solving and natural language comprehension in real-time distributed environments Experience based learning	Problem Space Hypothesis framework Working, semantic, and episodic memory – Relational Graph structure: Knowledge is represented as rule organized operators Procedural memory – Rule Iconic memory explicitly defined Reward system – appraisal based as well as user defined internal/external reward Specific modalities: Visual input Auditory Input	One decision at the time – single operator can be selected at each step, forcing a serial bottleneck Impasse – when knowledge about operator selection is insufficient or when an abstract operator cannot be implemented Goals are hierarchically organized The design of the perceptual-motor systems within SOAR is unrealistic, requiring users to define their own input and output functions for a given domain Approximate comparisons are hard and computationally intensive as implemented through graph-like representations

(*continued*)

Table 1. (*continued*)

	Approach	Learning algorithms	Characteristics	Weaknesses
				SOAR Agents are not endowed with general knowledge and only process ad-hoc built (or task-specific learned) symbolic knowledge structures
ACT-R (Adaptive Control of Thought – Rational)	Biologically based cognitive architecture to model human Behavior Knowledge and experience are represented using chunks and productions It is a hybrid paradigm architecture	Reinforcement learning for productions (linear discount version) Bayesian update for memory retrieval Production rules generation for learning of new representations	Supports semantic memory (encoded as chunks) and episodic memory Buffers encode working and episodic memory Specific modalities: Visual input Auditory Input Basic motor functions ACT-R allows to represent the information in terms of prototypes and exemplars and allow to perform, selectively, either prototype or exemplar-based categorization. This means that this architecture allows the modeler to manually specify which kind of categorization strategy to employ according to his specific needs	It cannot learn in real time It cannot learn from arbitrary stationary large and non-stationary databases – lack of adaptiveness to the environment Lack of goal prioritisation Lack of Adaptive heterogeneous fusion Lack of object feature search in an environment Does not assume a heterogeneous perspective – Cannot deal with conflicting information Cannot manage different reasoning strategies It is not able to autonomously decide which reasoning procedures to activate Task-specific knowledge and not with general cross-domain knowledge
NARS – Non-Axiomatic Reasoning System	Artificial General Intelligence (AGI) – software agent-based architecture Knowledge and experience are represented using beliefs, tasks, and concepts It is a symbolic architecture	Unified reasoning mechanism on a unified memory for learning,	Inference Engine Integrated Memory Control Mechanism Knowledge representation with an experience grounded semantics of the language, a set of inference rules with non-axiomatic logic support adaption in case of insufficient knowledge and resources The whole memory is semantic and incorporates cognitive maps	Task-specific knowledge and not with general cross-domain knowledge Lack of knowledge heterogeneity structures – lack of interaction between different reasoning strategies

(*continued*)

Table 1. (*continued*)

	Approach	Learning algorithms	Characteristics	Weaknesses
			Strong in reasoning with insufficient knowledge and resources It adopts a unified reasoning mechanism on a unified memory	
LIDA – Learning Intelligent Distribution Agent	Artificial General Intelligence (AGI) – software agent-based architecture Knowledge and experience are represented using perceptual knowledge – nodes and links in a Slipnet-like net with sensory data of various types attached to nodes Episodic knowledge – Boolean vectors (Sparse Distributed Memory Procedural knowledge – schemes a la Schema Mechanism It is an hybrid paradigm architecture	Constraint satisfaction Global Workspace Theory paradigms Reinforcement learning Global broadcasting	Cognitive cycle (action-perception cycle) acting as cognitive atom Cognitive cycles include sensory, perceptive, associative, workspace, transient episodic, declarative, procedural, global workspace, and sensory motor memory Distinct modules for perception, each of the listed memories, action selection, It can support control structures for software agents and robots It possesses an explicit attention mechanism [18, 19]	Task-specific knowledge and not with general cross-domain knowledge Lack of knowledge heterogeneity structures – lack of interaction between different reasoning strategies
ART (Adaptive Resonance Theory)	Biologically based cognitive architecture to model human Behavior Tries to model human memory and consciousness Knowledge and experience are represented using visual 3D boundary and surface representation It is a hybrid paradigm architecture	Non-linear neural networks with feedback – categorize events Unsupervised learning – categorization Supervise learning – anomaly detection Real-time learning – match learning and Vector Associative Map (VAM) Reinforcement learning – model how amygdala and basal ganglia interact with orbitofrontal cortex Bayesian effects as emergent properties Hebbian and non-Hebbian properties in learning dynamics Self-organizing maps with gradient descent – learning of new representations	Supports working memory (LTM Invariance Principle and Inhibition of Return rehearsal law), semantic memory (Limited association between chunks), episodic memory (limited spatial and temporal representations), procedural memory (multiple explicitly defined neural systems for learning, planning and control of action), iconic memory (emerges from role of top-down attentive interactions in laminar models of how the visual cortex sees), perceptual memory (model development of laminar visual cortex), cognitive map (neural networks), reward system (model how amygdala, hypothalamus, and basal ganglia interact with sensory and prefrontal cortex to learn to direct attention and actions towards valued goals), and attention control and consciousness	Computationally intensive Lack of knowledge heterogeneity structures – lack of interaction between different reasoning strategies

(*continued*)

Table 1. (*continued*)

	Approach	Learning algorithms	Characteristics	Weaknesses
			ART predicts a link between processes of Consciousness, Learning, Expectation, Attention, Resonance, and Synchrony (**CLEARS**) Learning from arbitrarily large databases The neural network enables local computations ART models can autonomously learn from non-stationary environments Memory stability – matches bottom up and top down representations Heterogeneous sensor fusion through multi-modal feature and hierarchical rule combinations Brain imaging studies	
CoJACK (Cognitive Java Agent Construction Kit)	Knowledge and experiences are represented using Beliefs-Desires-Intentions (BDI) architecture that handles events, plans and intentions (procedural memory), belief sets (declarative memory) and activation levels It has been used to model the variation in human Behavior It is a hybrid paradigm architecture	Reinforcement learning Bayesian Update BDI	It includes sets of beliefs for long term memory Procedural memory addresses events, plans, and intentions Event and goal manager are included It can function autonomously It uses ACT-R declarative memory equations It gets input from the world as events, which are then processed by plans	Time cost associated with adding beliefs or instantiating a plan Add noise to the decision-making process, which can affect the retrieval of beliefs and affect the selection of next intention to execute
ICARUS	Integrated cognitive architecture for physical agents with knowledge specified in the form of reactive skills, each denoting goal It is a symbolic architecture	Reinforcement learning State action pairs Belief desired and intention agents with high utility selection criteria Search trees	It includes several modules: Perceptual system Planning system Execution system Several memory systems Concepts are matched to percepts in a bottom-up way and goals are matched to skills in a top-down way Conceptual memory contains knowledge about general classes of objects and their relationships, while skill memory stores knowledge about the ways of doing things It has a Long-Term-Memory (LTM) and a Short-Term Memory (STM)	Lack of concurrent processing to cope with asynchronous inputs from multiple sensors while coordinating resources and actions across different modalities Uncertainty is not addressed

to dynamically and efficiently store and retrieve knowledge based on the experiences or events encountered by the system within the environment. Knowledge size refers to the dimension of the knowledge base available to the agents. Knowledge typology refers to theories on how humans organize, reason, and retrieve conceptual information. In the past, it was believed that concepts representation in the human brain was homogeneous and that concepts were categorized as classical, a prototype view, exemplar view, or theory-theoretical view. However, it is now believed that human may use in different instances different representation to categorize concepts, which led to the Heterogeneous Hypothesis about the nature of concepts. The heterogeneous hypothesis claims that different type of representation may exist and perhaps co-exist within the human brain. All such representations constitute different body of knowledge and contain different type of information associated with the same type of entity. Moreover, the heterogeneous hypothesis claims that each body of conceptual knowledge is distinguished by specific processes in which such representations are involved, such as in tasks like recognition, learning, categorization, planning, etc. The heterogeneous hypothesis, which assumes the availability of different types of knowledge encoded in a conceptual structure, is not implemented in any CA [20].

3 Cognitive Architecture: Evaluation Criteria

An additional challenge linked to the implementation of cognitive architectures is represented by the identification of key performance criteria to support their evaluation. Anderson and Lebeire [14] proposed to use the 12 criteria identified by Newell in 1980 to evaluate cognitive architectures on how well they do meet these functional criteria. These criteria represent an attempt for Newell to focus the field of cognitive architectures on the big picture needed to understand the human mind. They suggest calling the evaluation of the theory by this set of criteria "The Newell Test".

The criteria are reported in Table 2, first column. The first nine criteria reflect things that the architecture must achieve in order to implement human intellectual capability. The last three reflect constraints on how these functions must be achieved. In the second column of the table, the author has associated the artificial intelligence methodologies that could be used to assess a software agent architecture. In the third column, the author has created metrics associated to the functional criteria to objectively evaluate cognitive architectures criteria in a software agent platform.

The importance of considering quantitative evaluation criteria for cognitive architecture is to ensure that they can be applied across a range of engineering and industrial applications, rather than limited to studies on human cognition, behavior, and human performance. However, in the research the author carried out, cognitive architectures seem to be almost ignored in engineering autonomous vehicles application design.

Table 2. Newell's functional criteria for human cognitive architectures reported in Table 1 in [14] with associated AI technologies and Metrics for CA evaluation generated by the author.

Criteria	AI technologies to realize the functionality	Metrics to objectively evaluate CA
Functional criteria		
Flexible Behavior – Behave flexibly as a function of the environment	Learning Symbolic reasoning – It enables to perform arbitrary task to high level of expertise – cognitive plasticity Rule based systems	Percentage in Operational Performance Accuracy – Robustness in Operational changes in tasks Percentage of Errors in completing a task as it should Learning rate
Real-time performance – Operate in real-time	Neural network – parallel processing It becomes a constraint on learning as well as performance	Computational time to carry out a task
Adaptive Behavior – Exhibit rational, that is, effective adaptive Behavior – Does the system yield functional Behavior in the real world?	Decision Making – Decision trees and Utility function maximization	Percentage of action selection resulting in positive task completion Errors in action selected vs outcomes Robustness in Action Selection – Repeatability of right action selection to complete a task as it should – error in action selected can be considered as a limit that tends to zero
Vast Knowledge Base – Vast knowledge of the environment	Expert systems Big Data technologies, Data warehousing – Create heterogeneous classes of data to use as ground truth for computational functions	Minimum data required to perform tasks in a given environment – Assess Architecture performance in completing tasks vs. amount of data available and stored in databases Percentage of Errors in performing tasks vs richness of database Computational Memory capacity required by the system

(*continued*)

Table 2. (*continued*)

Criteria	AI technologies to realize the functionality	Metrics to objectively evaluate CA
Dynamic Behavior – Behave robustly in the face of error, the unexpected, and the unknown. The ability to deal with a dynamic and unpredictable environment is a precondition to survival for all organisms	Information fusion, Reasoning Decision making Probabilistic model – Bayes networks	Elapsed Time between reasoning and action selection to support states changes in response to external inputs Number of False positive and False Negatives
Knowledge Integration – Integrate diverse knowledge: Symbols and abstraction	Knowledge Base Systems Inference Utility functions Logic	Number of data classes in current databases – Update of knowledge databases with information heterogeneously sourced Computational Memory capacity required by the system
Natural Language	Symbolic reasoning Speech to text technologies Classification and clustering	Voice commands – Ability to control the system through natural language interaction
Consciousness – Exhibit self-awareness and a sense of self	Anomaly Detection Meta-reasoning Diagnostic Prognostic Forecasting	Percentage of Forecast Algorithms False positive and False Negatives – System health monitoring – diagnostic and prognostics Behavior Robustness (%) Behavior monitoring – feedback on actions vs. results – if the action produces the wanted result/task completion, then the behavior is good. Measure percentage of accomplishing the task Percentage of reasoning time per task completion – Feedback on reasoning – reasoning technologies assessed against situation picture Computational Memory capacity required by the system

(*continued*)

Table 2. (*continued*)

Criteria	AI technologies to realize the functionality	Metrics to objectively evaluate CA
Learning – Learn from its environment	Learning Short and Long-term memory logical design Inference engine Anomaly detection Rule based systems	Computational Memory capacity required by the system Rate of learning in the system
Functional constraints criteria		
Development – Acquire capabilities through development	Learning	Computational Memory capacity required by the system Percentage of Knowledge database reusability – Ability to adapt knowledge acquired to different scenarios
Evolution – Arise through evolution – Does the theory relate to evolutionary and comparative considerations?	Evolutionary Algorithms Reinforcement learning	Scalability – Software Functions self-building Maintainability – Software Functions self-upgrades
Brain – Be realizable within the brain: Do the components of the theory exhaustively map onto brain processes?	Integrated knowledge base algorithms	Overall system performance

4 Deep-Learning Artificial Neural Cognitive Architecture (D-LANCA)

The aim of this section is to address the current limits of cognitive architectures for autonomous systems specifically the issue of parallel execution of functionalities, heterogeneous knowledge representation, whilst improving computational efficiency, minimizing the limitations of historic knowledge database and knowledge limit, whilst improving real-time decision-making abilities. Specifically, the author is focusing on addressing the *heterogeneity* problem, which assumes the existence of multiple representations for a concept, and each concept represented by different kinds of categorization and reasoning mechanisms and processes are assumed to exist and require integration within a common framework. The objective is to create a common artificial general intelligent (AGI) cognitive architecture that can enable multi-functional processing and real time learning for autonomous driving systems.

The cognitive architecture is going to be developed through a modified deep learning neural network (DLNN) framework, which will enable the parallel processing of information and will be developed as a multi-agent systems (MAS) framework, interfacing with a knowledge base system represented by short and long-term memories, as well as

a goal management system to support indirect communication among nodes in case of lack of connections.

The rational to adopt a deep learning framework to implement a cognitive architecture resides in the ability of deep learning to support computational models with multiple process layers and learn representation of data with multiple level of abstraction [23].

Deep learning has been successfully applied in speech recognition, visual object recognition [23–29], object detection, drug discovery, genomics, etc. It enables the discovery of "intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep learning enables the computer to learn anything without human intervention and differs from traditional AI techniques as it enables learning though a hierarchical model in which each layer represents a level of abstraction for the problem versus the feature engineering, i.e. the manual definition of features within a data set. Also, Deep Learning works with unlabeled data and, most importantly, enables multiple decisions simultaneously.

The issue of *dimensionality* and *scalability*, as well as the knowledge size challenge previously described, will be addressed by automatically adding or removing nodes and connections within the network, or the self-programming characteristics of ANN (Rizk et al. 2019). This will enable to model the brain neuroplasticity. *Plasticity* can be advantageous to reduce computational time by reducing the search space to support decision making, actuation, and sensor selection, supporting efficiency in power management and goal management. Multiple goals can be passed to the system as an additional neural network layer. Plasticity can ease handling very large, hybrid, knowledge spaces and selection of actions to support dynamic system adaption and reaction in as real-time environment. The architecture has been named D-LANCA (Deep-Learning Artificial Neural Cognitive Architecture).

The cognitive process in D-LANCA is based on Boyd's OODA (Observe-Orient-Decide-Act) loop [31] and [32].

The need to associate a cognitive architecture to the OODA loop is funded on the need to associate operational and behavioral autonomous technologies within a unique framework.

The association of sensing functional elements within the observe class, the reasoning association within the orient class, and the decision making within the decide class of the OODA loop enable the autonomous technologies to be mapped onto an operational framework for the implementation of dynamic real-time autonomous platforms.

It is now important to create a link between the OODA loop and the Deep Learning Neural Network system as D-LANCA is designed as a modified deep neural network framework. The D-LANCA cognitive architecture is described in Fig. 1 and Fig. 2, in which the authors highlights the required changes.

Specifically, the modifications proposed to a conventional deep learning neural network framework are the following and captured in Fig. 3:

1. The propagation function in the neural layer ($\sum$) is no longer represented by a weighted sum of the input multiplied by their weights, but it becomes an operator node within the framework, whereby sensor fusion, information extraction and optimization, as well as the vehicle motion control can be implemented.

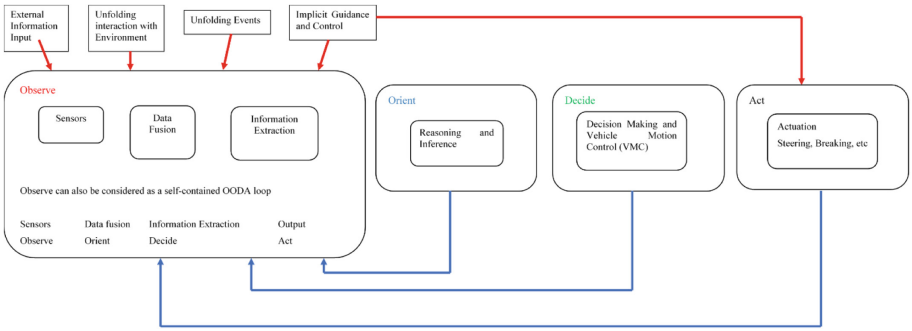


Fig. 1. D-LANCA linked to OODA loop

2. The weights modification in the network can be represented through the implementation principles of a spiking neural network (SNN), which will enable a bio-inspired learning through the weight modification based on the temporal and performance information provided by each modified activation function [27, 32–41]. By computing the weights through the temporal contribution of each activation function we can surpass the issues of training the SNN, as they are implemented through sums of Dirac delta functions that do not have derivatives to support a backpropagation algorithm implementation to test the network.

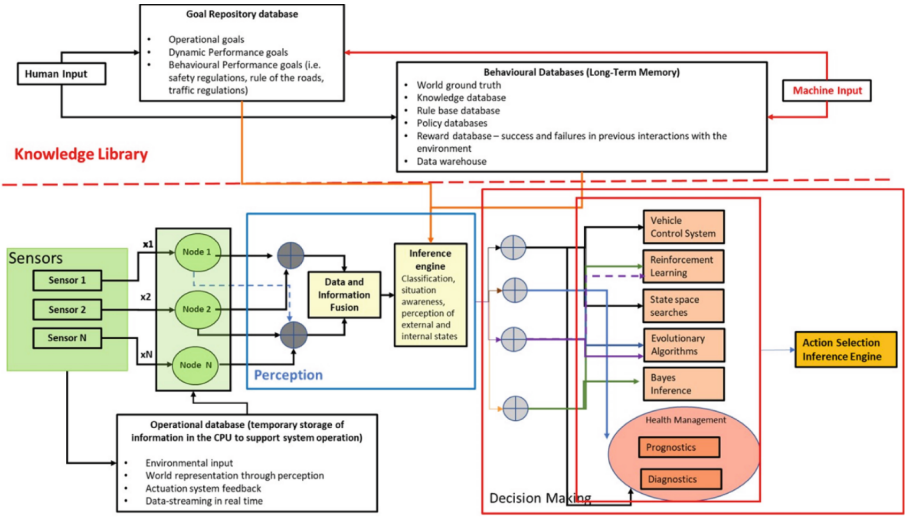


Fig. 2. D-LANCA architecture.

3. The activation function in the neural network is now considered as multi-agent software function, whereby computations such as information extraction, learning with for instance reinforcement learning (RL), decision making, etc. will be performed. By doing so, the software cognitive functionalities do not require to be executed

sequentially but they can be processed in parallel and the propagation function will output a multi-dimensional array to determine which functions need to be enabled.

4. The bias in the neural network are now considered inputs from knowledge databases, goals, human input, and world model created within the system.
5. As reported in [46], subjective estimation of the environment enables to create meaning within the system. However, subjective bias cannot be constants and they need to evolve as a function of the increased knowledge of the system. Therefore, as the databases are updated, the meaning of the semantic knowledge of the system will change and the bias will support the improvement in the overall inference of the system.
6. Learning, represented by the adaption of the network to better handle the task at hand within the given environment, is no longer solely represented by the adaption of the weights within the system, but by the expansion and contraction of the network through the inhibition of specific agents within the activation function as not relevant for the task, for instance, or the addition of new nodes to include additional information or inputs. This enables the implementation of plasticity as an embedded property within the system to learn the neural network structure and enable adaptive structures. In Russell and Norvig [30], one of the major problems in modelling a neural network is identified to be overfitting, which occurs when there are too many parameters in the model. Where when dealing with deep neural networks, one of the challenges in modelling the problem is represented by the number of hidden layers needed to support its solution and their sizes. A potential solution to plasticity is provided by Russell and Norvig ([30], p. 748) with the introduction of the optimal brain damage algorithm, which starts with a fully connected neural network and then removes connections from it, and with the tiling algorithm, which, on the other hand, starts with a single node network (perceptron) that tries to produce the correct output for as many example training sets as possible and adds subsequent nodes to handle the ones that could not be handled by the single perceptron.

The deep neural network is a recurrent neural network with the feedback loops supporting the implementation of cognition cycle.

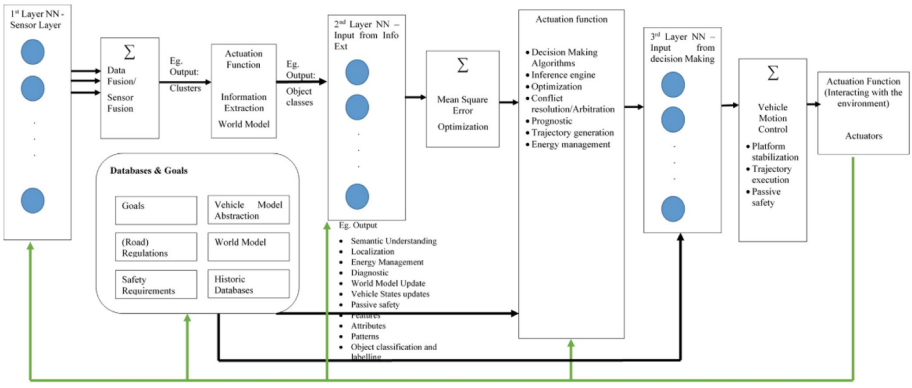


Fig. 3. D-LANCA explicit link to Deep Neural Network Architecture.

5 Conclusions and Further Work

In the work herein presented, the author has provided a systemic approach to support the design of a cognitive architecture able to support artificial general intelligence (AGI) implementation for an autonomous system.

An objective evaluation of commonly used cognitive architectures has been presented and metrics to evaluate their performance suggested.

The work has presented a deep neural network modified architecture as a cognitive architecture (D-LANCA). One of the advantages of using D-LANCA is the ability to support a robust representation of knowledge and support parallel functionalities to be co-processed in real time, mimicking the structure of a human brain. In addition, D-LANCA is envisaged to minimize the challenges of working in partially observable or not observable environment to support increased trust in the machine ability to learn and make decisions. In D-LANCA the author suggests to use the databases of historic data on behavior, environmental conditions, performance limits as a bias input within the system to support reasoning under uncertainty without limiting the learning and decision making ability of the machine

Further work is currently carried out to derive the mathematical model of D-LANCA and to implement a simulation model to test its performance against the derived evaluation criteria presented in Sect. 3. The author is focusing on the demonstrating the feasibility of changing the activation function into a MAS processing unit by implementing a reinforcement learning (RL) to create a trajectory planning function within the decision stage in the cognitive cycle. The rational for adopting RL as one of the functional building blocks in the activation function is that enables a fully data driven and self-learning model that does not rely on predictions, predefined rules or prior human knowledge [28]. This will address the challenge in AI of making good sequences of decisions under uncertainty. It will be assumed that information is readily available for the decision-making stage to be computed.

The applicability of this method will be demonstrated via modification of deep neural network focusing on the Decision Making/Thinking subsystem by deriving the equations to describe this subsystem and implementing a Matlab/Simulink modelling to support its analysis and evaluation.

References

1. Langley, P.: Cognitive architectures and general intelligent systems. AI Mag. **27**(2), 33–44 (2006)
2. Langley, P.: Information-processing psychology, artificial intelligence, and the cognitive systems paradigm thanks to. In: AAAI (2017)
3. Vernon, D., Metta, G., Sandini, G.: A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents. IEEE Trans. Evol. Comput. **11**(2), 151–180 (2007). https://doi.org/10.1109/TEVC.2006.890274
4. Models, C., Branch, A., Force, A., Patterson, W., Force, A.: Unified Theories of Cognition: Newell's Vision after 25 Years Presenters, pp. 250–251 (2012)
5. Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y.: An integrated theory of the mind. **111**(4), 1036–1060 (2004). https://doi.org/10.1037/0033295x.111.4.1036

6. Sun, R., Langley, P., Laird, J.E., Rogers, S.: Cognitive architectures: research issues and challenges. Cogn. Syst. Res. **10**(2), 141–160 (2009). https://doi.org/10.1016/j.cogsys.2006.07.004
7. Profanter, S.: Cognitive architectures (2012)
8. Lieto, A., Bhatt, M., Oltramari, A., Vernon, D.: The role of cognitive architectures in general artificial intelligence. Cogn. Syst. Res. **48**, 1–3 (2018). https://doi.org/10.1016/j.cogsys.2017.08.003
9. Duch, W., Oentaryo, R.J., Pasquier, M.: Cognitive architectures: where do we go from here? Front. Artif. Intell. Appl. **171**, 122–136 (2008)
10. Thagard, P.W.: Cognitive architectures. In: The Cambridge Handbook of Cognitive Science. Cambridge University Press, pp. 50–70 (2012)
11. Ritter, F.E.: Two cognitive modeling frontiers. Trans. Jpn. Soc. Artif. Intell. **24**, 241–249 (2009). https://doi.org/10.1527/tjsai.24.241
12. Kotseruba, I., Tsotsos, J.K.: A Review of 40 Years of Cognitive Architecture Research: Core Cognitive Abilities and Practical Applications (2016)
13. Ye, P., Wang, T., Wang, F.Y.: A survey of cognitive architectures in the past 20 years. IEEE Trans. Cybern. **48**(12), 3280–3290 (2018). https://doi.org/10.1109/TCYB.2018.2857704
14. Anderson, J.R., Lebiere, C.: The Newell Test for a Theory of cognition
15. Samsonovich, A.: Comparative Table of Cognitive Architectures (started on October 27, 2009; last update: June 18, 2012)
16. Samsonovich, A.V.: Comparative analysis of implemented cognitive architectures. Front. Artif. Intell. Appl. **233**, 469–479 (2011). https://doi.org/10.3233/978-1-60750-959-2-469
17. Kingdon, R.: A review of cognitive architectures. ISO Project report (2008)
18. Franklin, S., Madl, T., D'Mello, S., Snaider, J.: LIDA: a systems-level architecture for cognition, emotion, and learning. IEEE Trans. Auton. Ment. Dev. **6**(1), 19–41 (2014). https://doi.org/10.1109/TAMD.2013.2277589
19. Computing, C.: The Mind According to LIDA - A Brief account The "LIDA Model" and its Cognitive Cycle, pp. 1–20 (2013)
20. Lieto, A., Lebiere, C., Oltramari, A.: The knowledge level in cognitive architectures: current limitations and possible developments. Cogn. Syst. Res. **48**, 39–55 (2018). https://doi.org/10.1016/j.cogsys.2017.05.001
21. Li, D.: A tutorial survey of architectures, algorithms. APSIPA Trans. Signal Inf. Process. **3**(2014), 1–29 (2014)
22. Lieto, A.: Representational limits in cognitive architectures. CEUR Workshop Proceedings, vol. 1855, pp. 16–20 (2017)
23. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
24. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: Neural networks architectures review. 1–31 (2017)
25. Liu, Y., Xiang, C.: Hybrid learning network: a novel architecture for fast learning. Procedia Comput. Sci. **122**, 622–628 (2017)
26. Luo, X., Shen, R., Hu, J., Deng, J., Hu, L., Guan, Q.: A deep convolution neural network model for vehicle recognition and face recognition. Procedia Comput. Sci. **107**(ICICT), 715–720 (2017)
27. Petersen, S.E., Sporns, O.: Brain networks and cognitive architectures. Neuron **88**(1), 207–219 (2015)
28. Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., Barth, M.: Deep reinforcement learning enabled self-learning control for energy efficient driving. Transp. Res. Part C Emerging Technol. **99**, 67–81 (2019)
29. Rizk, Y., Hajj, N., Mitri, N., Awad, M.: Deep belief networks and cortical algorithms: a comparative study for supervised classification. Appl. Comput. Inf. **15**(2), 81–93 (2019)

30. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach, vol. 9 (1995)
31. Behere, S., Törngren, M., Chen, D.: A reference architecture for cooperative driving. J. Syst. Architect. **59**(10), 1095–1112 (2013)
32. Brehmer, B.: The dynamic OODA loop: amalgamating Boyd's OODA loop and the cybernetic approach to command and control. In: 10th International Command and Control Research and Technology Symposium The Future of C2 (2005)
33. Huyck, C.R.: A neural cognitive architecture. Cogn. Syst. Res. **59**, 171–178 (2020)
34. Kim, J., Kim, H., Huh, S., Lee, J., Choi, K.: Deep neural networks with weighted spikes. Neurocomputing **311**, 373–386 (2018)
35. Sboev, A., Vlasov, D., Rybka, R., Serenko, A.: Spiking neural network reinforcement learning method based on temporal coding and STDP. Procedia Comput. Sci. **145**, 458–463 (2018)
36. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T., Maida, A.: Deep learning in spiking neural networks. Neural Netw. **111**, 47–63 (2019)
37. Wu, X., Wang, Y., Tang, H., Yan, R.: A structure-time parallel implementation of spike-based deep learning. Neural Netw. **113**, 72–78 (2019)
38. Wang, B., Chen, L.L., Zhang, Z.Y.: A novel method on the edge detection of infrared image. Optik **180**, 610–614 (2019)
39. Stief, P., Dantan, J.-Y., Etienne, A., Siadat, A.: A New Methodology to Analyze the Functional and Physical Architecture of Existing Products for an Assembly Oriented Product Family Identification (2018)
40. Seijen, V., Harm, M.F., Romoff, J., Laroche, R., Barnes, T., Tsang, J.: Hybrid reward architecture for reinforcement learning. In Advances in Neural Information Processing Systems 2017 (NIPS 2017), pp. 5393–5403 (2017)
41. Qi, X., Luo, Y., Wu, G., Boriboonsomsin, K., Barth, M.: Deep reinforcement learning enabled self-learning control for energy efficient driving. Transp. Res. Part C Emerging Technol. **99**, 67–81 (2019)

Learn-Able Parameter Guided Activation Functions

S. Balaji[1], T. Kavya[1(✉)], and Natasha Sebastian[2]

[1] HCL Technologies Ltd., Chennai, India
{balaji-sba,kavya-t}@hcl.com
[2] Delhi Technological University, Delhi, India
https://www.hcltech.com
http://dtu.ac.in/

Abstract. In this paper, the concept of adding learn-able slope and mean shift parameters to an activation function to improve the total response region is explored. The characteristics of an activation function depend highly on the value of parameters. Making the parameters learn-able, makes the activation function more dynamic and capable to adapt as per the requirements of it's neighboring layers. The introduced slope parameter is independent of other parameters in the activation function. The concept was applied to ReLU to develop Dual Line and Dual Parametric ReLU activation function. Evaluation on MNIST and CIFAR10 show that the proposed activation function Dual Line achieves top-5 position for mean accuracy among 43 activation functions tested with LENET4, LENET5 and WideResNet architectures. This is the first time more than 40 activation functions were analyzed on MNIST and CIFAR10 dataset at the same time. The study on the distribution of positive slope parameter β indicates that the activation function adapts as per the requirements of the neighboring layers. The study shows that model performance increases with the proposed activation functions.

Keywords: Activation function · Dual line · DP ReLU

1 Introduction

The activation functions used across the layers of deep neural networks play a significant role in the ability of the whole network to achieve good performance. Though each layer of a deep neural network can have different requirements, the convention is to use the same activation function at each output of a layer. Therefore using a good activation function suitable for every output node in the layer is essential.

Rectified Linear Units (ReLU) [14] and its variants such as Leaky ReLU [11], Parametric ReLU (PReLU) [6] are the most commonly used activation functions due to their simplicity and computational efficiency. The introduction of learn-able parameter in the negative axis for PReLU increased the overall response region. Though extra parameters are introduced in PReLU activation

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 583–597, 2021.
https://doi.org/10.1007/978-3-030-55180-3_43

function, the number of parameters added due to PReLU is negligible compared to the total number of parameters in the network. Another activation function that makes use of learn-able parameters is Parametric ELU (PELU) [23]. Learn-able parameters in PELU activation function adopted characteristics as per the requirements of the training stage. In the case of PELU, the positive axis parameter is dependent as it is defined as the ratio of parameters used to alter exponential decay and saturation point. In this paper, two activation functions are proposed, with a positive slope parameter which is independent of other parameters, allowing it to dynamically update.

In Flexible ReLU [18] and General ReLU, mean shift parameters were introduced to shift the mean activation close to zero. The Dual Line activation function can be viewed as a combination of DP ReLU with learn-able mean shift parameter. The better performance of Dual Line compared to DP ReLU clearly indicates the impact of the mean shift parameter.

The rest of the paper is organized as follows. Section 2 describes our proposed activation function and Sect. 3 deals with their properties. Section 4 describes the steps to be carried out to extend the proposed concept to other activation functions. Experimental analysis and performance evaluation are described in Sect. 5 and Sect. 6, respectively. Results and discussion are detailed in Sect. 7 and Sect. 8 respectively. The paper is concluded in Sect. 9.

2 Proposed Activation Functions

2.1 Dual Parametric ReLU (DP ReLU)

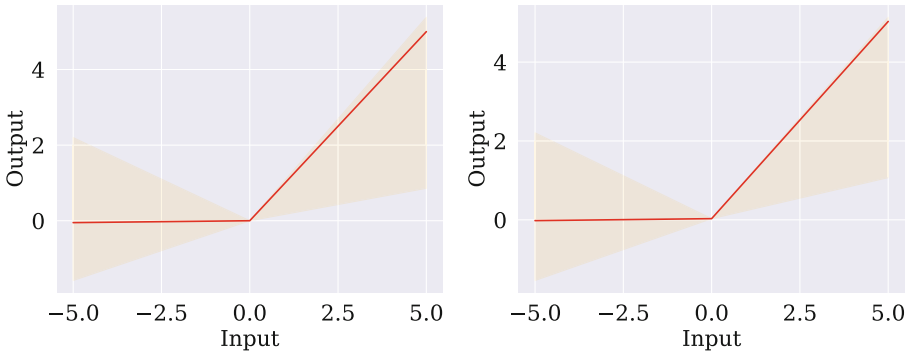
DP ReLU is a new variant of ReLU with a learn-able slope parameter in both axes. The difference between Parametric ReLU (PReLU) and DP ReLU is the usage of learn-able slope parameter in the positive axis. The slope parameters α and β are initialized with 0.01 and 1 respectively as in PReLU and Leaky ReLU. For slope parameters (α and β), DP ReLU is defined as

$$X = \begin{cases} \alpha \times x, & \text{if } x < 0 \ . \\ \beta \times x, & \text{if } x > 0 \ . \end{cases} \tag{1}$$

2.2 Dual Line

Dual Line is an extension of the DP ReLU. Learn-able slope parameters are multiplied to both axes and the mean shift parameter is added. The resultant activation function resembles the line equation in both axes. Mean parameter is initialized with a value of -0.22 by adding the mean shift (-0.25) and threshold (0.03) parameters used in TReLU[10]. For slope parameters (α and β) and mean shift parameter (m), Dual Line is defined as

$$X = \begin{cases} \alpha \times x + m, & \text{if } x < 0 \ . \\ \beta \times x + m, & \text{if } x > 0 \ . \end{cases} \tag{2}$$



(a) DP ReLU Activation Function (b) Dual line Activation Function

Fig. 1. Plot of DP ReLU and Dual Line activation function for different values of learn-able parameter. The values were obtained from WideResNet models trained on CIFAR10. *Red line* indicates the default initialization state. The *filled region* indicates the overall response region of the activation function, which is obtained by finding the min and max response curves observed across the network for the activation function

3 Properties of DP ReLU and Dual Line

3.1 Independent

Both the slope parameters are independent of other parameters and act directly on the input without any constraints as shown in Eq. (1) and Eq. (2).

3.2 Large Response Region

As shown in Fig. 1, the learn-able parameters can take different values, so the proposed activation function has a larger response region compared to the variants without learn-able parameters.

3.3 Slope Parameter in Positive Axis

The value of $\beta > \text{x}$ results in boosting the activation and $\beta < \text{x}$ results in attenuation of the activation. The final value of β in the model depends on the position of the activation function with respect to other layers.

3.4 Mean Shifting Due to Mean Shift Parameter

As per Fisher optimal learning criteria, the undesired bias shift effect can be reduced by centering the activation at zero or by using activation with negative values [1]. Unit natural gradient can be achieved by pushing mean activation close to zero. This reduces the risk of over-fitting and allows the network to learn faster [2,4]. The mean shift parameter in Dual Line activation function aids to push the mean activation towards zero.

3.5 Computation Requirements

Using complex mathematical operations in the activation function increases compute time and memory requirement for training and inference. The absence of exponential or division makes ReLU and its variants faster [16]. ISRLU uses inverse square roots instead of exponentials as they exhibit 1.2X better performance in Intel Xeon E5-2699 v3 (Haswell AVX2) [2]. The proposed activation functions does not have complex mathematical operations. The only bottleneck in compute time during training is due to the inclusion of learn-able parameters.

4 Extending the Concept to Other Activation Functions

Most activation functions have an unbounded near-linear response in the Ist quadrant. The concept of adding learn-able parameter in the positive axis and mean shift parameter can be extended to other activation functions.

An existing activation function (G) can be modified by treating it as a piecewise function and replacing the characteristics for x > 0. For value of x > 0, the function can be defined as input multiplied by learn-able slope parameter and it remains the same elsewhere.

For an activation function G defined as follows:

$$X = G(x) \ . \tag{3}$$

The proposed concept can be applied as follows:

$$X = \begin{cases} G(x) + m, & \text{if } x < 0. \\ \beta \times x + m, & \text{if } x > 0. \end{cases} \tag{4}$$

5 Experimental Analysis

5.1 Data Analysis

MNIST [13], Fashion MNIST [5], CIFAR10 [3], CIFAR100 [3], ImageNet 2012 [8], Tiny ImageNet [22], LFW [9], SVHN [21], and NDSB [15] are the computer vision datasets used for analyzing activation functions. Experiments are carried out with MNIST and CIFAR10 in this paper, as they are the most frequently used datasets.

5.2 Experimental Setup

PyTorch deep-learning library was used for the experiments [17]. Adam optimizer and Flattened cross-entropy loss are used. Learning rate was estimated using learning rate finder [19]. The max and min values of learning rate across multiple runs are presented, which can be an indicator for the range of values the model prefers.

With hyper-parameters and all other layers kept constant, the activation function was replaced to analyze which activation function aids the network to learn faster and achieve better accuracy in minimal epochs. For each activation function, five iterations on each of the datasets were done. Computational speedup required for analyzing 43 activation functions was achieved using mixed-precision training [12].

5.3 MNIST - LENET5 and LENET4

LENET5 comprises of 2 convolutional layers followed by 3 linear layers. LENET4 comprises of 2 convolutional layers followed by 2 linear layers. Both LENET networks do not have batch normalization layers.

5.4 CIFAR10 - WideResNet

The hyper-parameters were based on fast.ai submission for the DAWNBench challenge [20]. The normalized data were flipped, and random padding was carried out. The training was carried out with 512 as batch size for 24 epochs and learning rate estimated as per the learning rate estimator. Mixup data augmentation was carried out on the data [24].

6 Performance Evaluation

Metrics such as accuracy, top-5 accuracy, validation loss, training loss and time are estimated for the 3 networks. WideResNet contains batch norm layers and the LENET network does not. The impact of batch normalization layer would be one of the factors to consider between these architectures. The main analysis parameter is mean accuracy across 5 runs.

7 Results

The following sections discuss the results and analysis of training LENET5 and LENET4 on the MNIST dataset and WideResNet on the CIFAR10 dataset.

7.1 MNIST LENET5

Dual Line achieves the 2nd best accuracy and best mean accuracy. DP ReLU achieves 18th and 19th in accuracy and mean accuracy respectively. Top accuracy is observed in GELU [7] Dual Line achieves 2nd and 3rd rank w.r.t mean train and validation loss. DP ReLU achieves 18th and 20th in mean train and validation loss respectively.

NAME	Loss		Mean Loss		Acc	Mean Acc	Learning Rate		Time (s)
	Train	Valid	Train	Valid			Min	Max	
Dual Line	0.4943	0.0542	0.5002	0.0580	0.9901	0.9895	0.0025	0.0275	4.03
SELU	0.5079	0.0609	0.5125	0.0642	0.9900	0.9894	0.0025	0.0030	3.37
CELU	0.5019	0.0575	0.5049	0.0597	0.9896	0.9890	0.0025	0.0191	3.60
PoLU	0.5047	0.0600	0.5085	0.0641	0.9897	0.9889	0.0030	0.0036	3.87
ISRLU	0.5005	0.0584	0.5052	0.0595	0.9898	0.9888	0.0025	0.0229	3.67
PELU	0.4931	0.0513	0.4983	0.0567	0.9897	0.9885	0.0025	0.0025	4.00
SiLU	0.5010	0.0538	0.5061	0.0593	0.9894	0.9885	0.0025	0.0191	3.30
ELU	0.5014	0.0600	0.5079	0.0617	0.9892	0.9884	0.0025	0.0030	3.33
Mish	0.5043	0.0502	0.5117	0.0585	0.9901	0.9884	0.0025	0.0229	3.93
Swish	0.5038	0.0561	0.5113	0.0621	0.9891	0.9883	0.0025	0.0229	3.60
PReLUC	0.5136	0.0619	0.5156	0.0653	0.9896	0.9883	0.0030	0.0191	3.67
PReLU(0.01)	0.5101	0.0532	0.5156	0.0641	0.9897	0.9881	0.0030	0.0191	3.90
LiSHT	0.4930	0.0499	0.5116	0.0565	0.9888	0.9881	0.0025	0.0076	3.67
ELiSH	0.5118	0.0567	0.5166	0.0617	0.9897	0.9880	0.0025	0.0275	3.73
PReLU	0.5107	0.0631	0.5169	0.0657	0.9889	0.9879	0.0025	0.0191	3.47
LeakyReLU	0.5147	0.0640	0.5206	0.0679	0.9886	0.9879	0.0025	0.0030	3.30
GELU	0.5127	0.0541	0.5142	0.0612	0.9905	0.9878	0.0030	0.0229	3.87
TReLU	0.5155	0.0567	0.5207	0.0642	0.9888	0.9877	0.0025	0.0229	3.97
DP ReLU	0.5102	0.0574	0.5183	0.0663	0.9886	0.9875	0.0025	0.0030	4.00
RReLU	0.5263	0.0590	0.5295	0.0643	0.9886	0.9875	0.0030	0.0229	3.30
ARiA2	0.5207	0.0591	0.5249	0.0652	0.9876	0.9872	0.0025	0.0275	3.83
BentIdentity	0.5368	0.0702	0.5456	0.0756	0.9880	0.9872	0.0012	0.0021	3.63
ReLU6	0.5243	0.0657	0.5295	0.0721	0.9872	0.9868	0.0030	0.0275	3.60
SQNL	0.5240	0.0676	0.5262	0.0735	0.9886	0.9866	0.0052	0.0076	3.87
LeakyReLU(0.01)	0.5221	0.0621	0.5302	0.0705	0.9874	0.9865	0.0030	0.0191	3.77
FTSwishPlus	0.5058	0.0609	0.5172	0.0692	0.9878	0.9865	0.0030	0.0275	3.77
RationalTanh	0.5243	0.0724	0.5307	0.0815	0.9874	0.9859	0.0036	0.0052	3.70
ReLU	0.5244	0.0671	0.5335	0.0737	0.9870	0.9859	0.0030	0.0191	3.37
GReLU	0.5293	0.0707	0.5357	0.0731	0.9867	0.9855	0.0030	0.0229	3.67
Tanh	0.5201	0.0695	0.5298	0.0762	0.9862	0.9853	0.0052	0.0132	3.70
Atan	0.5335	0.0739	0.5383	0.0788	0.9859	0.9852	0.0063	0.0076	3.43
HardTanh	0.5296	0.0695	0.5382	0.0772	0.9872	0.9852	0.0030	0.0044	3.47
ISRU	0.5249	0.0736	0.5295	0.0795	0.9865	0.9851	0.0110	0.0158	3.70
Softsign	0.5406	0.0803	0.5456	0.0872	0.9852	0.9848	0.0110	0.0191	3.73
BReLU	0.5365	0.0721	0.5450	0.0756	0.9857	0.9845	0.0036	0.0275	3.97
RectifiedTanh	0.5439	0.0772	0.5492	0.0855	0.9840	0.9827	0.0132	0.0275	3.63
dSiLU	0.5799	0.1001	0.5848	0.1049	0.9837	0.9824	0.0331	0.1738	3.77
HardSigmoid	0.5646	0.0944	0.5823	0.1001	0.9832	0.9818	0.0275	0.0479	3.70
Tanhshrink	0.5799	0.0746	0.5905	0.0840	0.9826	0.9813	0.0063	0.0191	3.40
LogSigmoid	0.5923	0.0934	0.5964	0.1004	0.9794	0.9787	0.0132	0.1738	3.47
HardShrink	0.6253	0.1031	0.6325	0.1073	0.9769	0.9765	0.0025	0.0191	3.37
Softshrink	0.6105	0.0923	1.9630	1.8593	0.9804	0.2869	0.0000	0.1445	3.60
ThresholdedReLU	2.3010	2.3010	2.3011	2.3010	0.1135	0.1135	0.0000	0.0000	3.67
	Low								High

Fig. 2. Results for LENET5 network with different activation functions trained on the MNIST dataset. The lite to dark transition corresponds to low to high values. For loss and time, low values are preferred. For accuracy, high values are preferred. Time refers to average training time per epoch in seconds

Name	Loss		Mean Loss		Acc	Mean	Learning Rate		Time
	Train	Valid	Train	Valid		Acc	Min	Max	(s)
PELU	0.7169	0.1791	0.7241	0.1848	0.9627	0.9618	0.0191	0.0331	4.00
LiSHT	0.8144	0.2871	0.8292	0.3051	0.9477	0.9436	0.0331	0.0832	3.95
BentIdentity	0.7807	0.2483	0.7855	0.2545	0.9449	0.9417	0.0479	0.0832	3.90
Dual Line	0.8415	0.3101	0.8517	0.3231	0.929	0.9242	0.0832	0.3631	4.00
SELU	0.8395	0.3402	0.8454	0.3457	0.9247	0.9212	0.0832	0.1738	3.60
DP ReLU	0.8384	0.3127	0.8651	0.3379	0.9274	0.9191	0.0575	0.2512	4.00
PoLU	0.8747	0.3665	0.8817	0.3744	0.9158	0.9128	0.1445	0.1738	4.00
ISRLU	0.8855	0.3710	0.8925	0.3788	0.9118	0.9106	0.1202	0.1738	4.00
CELU	0.8782	0.3683	0.8857	0.3769	0.9114	0.9104	0.1202	0.1738	3.55
ELU	0.8742	0.3648	0.8879	0.3783	0.9163	0.9093	0.1202	0.2512	3.55
GELU	0.8942	0.3810	0.9027	0.3892	0.9108	0.9091	0.1738	0.3631	4.00
Mish	0.8942	0.3847	0.9056	0.3923	0.9099	0.9074	0.1202	0.6310	4.00
ARiA2	0.9027	0.3845	0.9154	0.4000	0.9105	0.9072	0.0692	0.3631	4.00
PReLU	0.9035	0.3871	0.9113	0.3935	0.9079	0.9054	0.0692	0.3631	3.40
PReLUC	0.8965	0.3798	0.9071	0.3932	0.9094	0.9049	0.1202	0.2089	3.45
LeakyReLU(0.01)	0.9103	0.3935	0.9151	0.4011	0.9073	0.9038	0.0832	0.2512	4.00
ELiSH	0.9135	0.3979	0.9309	0.4082	0.9084	0.9035	0.1000	0.3631	3.85
RReLU	0.8940	0.3757	0.9203	0.4004	0.9098	0.9028	0.1445	0.2089	3.45
TReLU	0.9145	0.3997	0.9285	0.4105	0.9048	0.9027	0.1000	0.3631	4.00
LeakyReLU	0.9104	0.3907	0.9232	0.4113	0.9108	0.9020	0.1445	0.2512	3.45
GReLU	0.8977	0.3861	0.9195	0.4070	0.9086	0.9015	0.1202	0.2512	4.00
HardShrink	0.8914	0.3853	0.9013	0.3943	0.9036	0.9013	0.0692	0.2089	3.30
ReLU	0.8943	0.3775	0.9306	0.4143	0.9124	0.9013	0.0021	1.9055	3.65
Swish	0.9155	0.3943	0.9262	0.4091	0.9059	0.9012	0.0832	0.3631	3.85
FTSwishPlus	0.9160	0.3965	0.9368	0.4165	0.9051	0.9009	0.1000	0.2089	4.00
PReLU(0.01)	0.9075	0.3910	0.9286	0.4133	0.9088	0.9007	0.0832	0.2512	4.00
SiLU	0.9037	0.3927	0.9362	0.4144	0.9084	0.9006	0.1000	0.2512	3.60
ReLU6	0.9293	0.4298	0.9660	0.4796	0.9096	0.8962	0.0692	1.9055	3.45
HardTanh	0.9844	0.5317	0.9935	0.5401	0.8801	0.8776	0.0832	0.1738	3.40
BReLU	0.9865	0.4770	1.0585	0.5346	0.8844	0.8706	0.1202	0.5248	3.90
RationalTanh	1.0322	0.5980	1.0398	0.6059	0.8703	0.8670	0.1000	0.3020	3.70
Softshrink	0.9069	0.3807	1.2206	0.7472	0.9069	0.8124	0.1445	0.3631	3.50
Atan	1.3547	0.9845	1.3723	1.0030	0.8091	0.8030	0.3020	0.3631	3.75
Tanh	1.4497	1.1027	1.5161	1.1816	0.7933	0.7725	0.3631	0.4365	3.60
SQNL	1.5591	1.2356	1.5823	1.2639	0.7738	0.7598	0.3631	0.5248	4.00
ISRU	1.9906	1.8473	2.0332	1.9064	0.7035	0.6752	0.7586	0.7586	3.95
Softsign	2.0647	1.9518	2.0995	2.0019	0.6998	0.6674	0.7586	1.0965	3.75
RectifiedTanh	2.1375	2.0648	2.1748	2.1170	0.7155	0.6346	0.4365	0.9120	3.75
ThresholdedReLU	1.5965	1.0087	1.8945	1.5258	0.7938	0.5504	0.2089	0.3631	3.70
LogSigmoid	2.1802	2.0915	2.2231	2.1712	0.5505	0.4447	0.7586	1.0965	3.30
HardSigmoid	2.2794	2.2697	2.2839	2.2755	0.2098	0.1737	0.0000	0.7586	3.60
dSiLU	2.2823	2.2742	2.2856	2.2783	0.2042	0.1522	0.7586	1.5849	3.90
Tanhshrink	2.3021	2.3017	2.3024	2.3022	0.1203	0.1061	0.0000	3.3113	3.45
	Low								High

Fig. 3. Results for LENET4 network with different activation functions trained on the MNIST dataset. The lite to dark transition corresponds to low to high values. For loss and time, low values are preferred. For accuracy, high values are preferred. Time refers to average training time per epoch in seconds

Name	Loss		MeanLoss		Acc	Mean Acc	LearningRate		Time (s)
	Train	Valid	Train	Valid			Min	Max	
ARiA2	0.6730	0.2046	0.6772	0.2103	0.9459	0.9435	0.0021	0.0044	31.44
DualLine	0.6549	0.2025	0.6599	0.2139	0.9451	0.9422	0.0014	0.0036	35.03
FTSwishPlus	0.6785	0.2038	0.6825	0.2111	0.9451	0.9418	0.0025	0.0091	29.52
BReLU	0.6792	0.2026	0.6872	0.2153	0.9461	0.9417	0.0017	0.0052	26.52
PReLU(0.01)	0.6634	0.2095	0.6699	0.2183	0.9429	0.9405	0.0017	0.0025	25.80
ELiSH	0.6933	0.2137	0.6990	0.2210	0.9433	0.9403	0.0017	0.0030	38.08
GELU	0.6815	0.2053	0.6988	0.2249	0.9462	0.9384	0.0010	0.0036	36.73
ReLU6	0.6718	0.2105	0.6923	0.2235	0.9419	0.9384	0.0017	0.0052	24.09
DPReLU	0.6539	0.2024	0.6767	0.2242	0.945	0.9382	0.0010	0.0036	34.17
GReLU	0.6801	0.2068	0.6987	0.2265	0.9443	0.9378	0.0014	0.0044	24.04
PReLU	0.6578	0.2065	0.6869	0.2280	0.9444	0.9369	0.0008	0.0030	25.51
TReLU	0.6869	0.2147	0.7006	0.2273	0.9411	0.9369	0.0014	0.0030	25.02
LeakyReLU(0.01)	0.6795	0.2088	0.7042	0.2309	0.9434	0.9366	0.0010	0.0044	24.65
SiLU	0.7196	0.2224	0.7263	0.2346	0.9416	0.9362	0.0014	0.0044	27.01
Swish	0.7203	0.2288	0.7250	0.2348	0.9375	0.9354	0.0014	0.0063	28.00
ReLU	0.6900	0.2246	0.7021	0.2334	0.9389	0.9347	0.0014	0.0025	24.08
LeakyReLU	0.7039	0.2294	0.7210	0.2413	0.9362	0.9324	0.0010	0.0030	24.01
Mish	0.7128	0.2252	0.7337	0.2469	0.9377	0.9308	0.0010	0.0036	29.00
PELU	0.6723	0.2117	0.7233	0.2488	0.942	0.9301	0.0003	0.0036	42.18
RReLU	0.7469	0.2529	0.7806	0.2817	0.9298	0.9192	0.0007	0.0025	26.00
ISRLU	0.7812	0.2814	0.7921	0.2902	0.9213	0.9178	0.0010	0.0014	40.20
PReLUC	0.7559	0.2662	0.7941	0.3008	0.9229	0.9126	0.0006	0.0010	25.00
RectifiedTanh	0.7340	0.2530	0.7886	0.3046	0.9289	0.9096	0.0012	0.0036	26.00
LogSigmoid	0.8159	0.3091	0.8435	0.3379	0.9129	0.9027	0.0005	0.0025	25.00
CELU	0.8654	0.3565	0.9107	0.4002	0.8959	0.8800	0.0004	0.0010	24.31
ELU	0.8945	0.3837	0.9479	0.4448	0.887	0.8651	0.0002	0.0007	24.08
PoLU	0.8439	0.3334	0.9512	0.4611	0.9023	0.8580	0.0001	0.0014	33.72
BentIdentity	0.9018	0.3904	0.9952	0.4948	0.8847	0.8475	0.0001	0.0010	30.99
LiSHT	0.7050	0.2584	0.9193	0.4894	0.9281	0.8444	0.0004	0.0036	30.50
SELU	0.9964	0.5017	1.1347	0.6773	0.8449	0.7820	0.0001	0.0008	24.18
Softsign	0.9965	0.5185	1.2123	0.7969	0.8393	0.7377	0.0001	0.0017	29.20
Softshrink	1.0440	0.5593	1.2974	0.8722	0.8261	0.7061	0.0001	0.0010	24.11
ThresholdedReLU	1.2577	0.8325	1.3098	0.9247	0.7282	0.6980	0.0036	0.0110	24.00
ISRU	1.1802	0.7664	1.2999	0.9147	0.7487	0.6944	0.0002	0.0006	35.00
SQNL	1.1117	0.6634	1.3628	0.9915	0.7881	0.6636	0.0001	0.0007	39.82
Tanh	1.3810	1.0148	1.4242	1.0742	0.6558	0.6333	0.0001	0.0002	24.37
Atan	1.3802	1.0106	1.4689	1.1309	0.6565	0.6126	0.0001	0.0002	25.03
RationalTanh	1.4147	1.0576	1.5336	1.2174	0.6389	0.5813	0.0001	0.0002	26.00
HardShrink	1.4588	1.0938	1.5582	1.2312	0.6223	0.5747	0.0001	0.0002	24.03
HardTanh	1.6641	1.3785	1.6785	1.3980	0.517	0.5116	0.0000	0.0001	24.08
Tanhshrink	1.4517	1.0636	1.7615	1.7069	0.638	0.3812	0.0000	0.0479	26.14
dSiLU	1.2013	0.7770	1.9105	1.7212	0.7456	0.3508	0.0002	3.3113	31.75
HardSigmoid	1.1086	0.6637	1.8648	1.9718	0.7913	0.2618	0.0005	2.2909	27.52
	Low								High

Fig. 4. Results for WideResNet with different activation functions trained on the CIFAR10 dataset. The lite to dark transition corresponds to low to high values. For loss and time, low values are preferred. For accuracy, high values are preferred. Time refers to average training time per epoch in seconds

7.2 MNIST LENET4

Dual Line secure 4th and 5th rank w.r.t accuracy and mean accuracy. DP ReLU secures 5th and 6th position in accuracy and mean accuracy. PELU achieves the best performance in each of the metrics.

Dual Line achieves 5th and 4th in mean train and validation loss. DP ReLU secures 6th and 5th in mean train and mean validation loss.

7.3 CIFAR10 WideResNet

DP ReLU and Dual Line achieve 9th and 2nd best mean accuracy. Aria2 achieves the best mean accuracy of 0.9435. The highest accuracy value was observed with GELU. Dual Line and DP ReLU achieve 4th and 6th best accuracy. Best mean top-5 accuracy is observed in General ReLU. Dual Line and DP ReLU achieve 5th and 15th in mean top-5 accuracy.

DP ReLU and Dual Line achieve the best and 2nd best in Train and Validation loss. Dual Line and DP ReLU secure best and 3rd best mean train loss. 3rd and 8th best in validation loss for Dual Line and DP ReLU respectively. DP ReLU performance decrease as the number of linear layers in the model increase, which can be seen by comparing its performance in LENET4 and LENET5 models.

8 Discussion

8.1 Learning Rate Analysis

The learning rate estimated varies w.r.t activation used. Learning rates were estimated to indicate the range of values an activation function prefers for a dataset and are shown in the Fig. 2, Fig. 3 and Fig. 4.

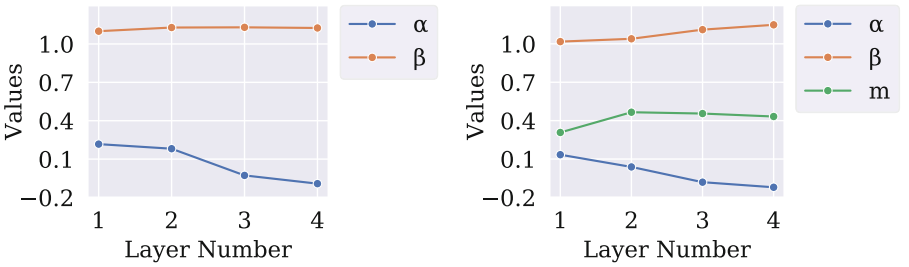
Table 1. Analysis of learning rate values observed for each of the datasets across 5 runs. The 'Overall' row indicates the overall maximum and minimum value observed for a dataset and name of the corresponding activation function

Activation function	LENET5		LENET4		WideResNet	
	Min	Max	Min	Max	Min	Max
DP Relu	2.5E−03	3.0E−03	5.7E−02	2.5E−01	1.0E−03	3.6E−03
Dual Line	2.5E−03	2.7E−02	8.3E−02	3.6E−01	1.4E−03	3.6E−03
Overall	1.0E−06	1.7E−01	1.0E−06	3.3E+00	1.0E−06	3.3E+00
	ThresholdedReLU	LogSigmoid	Tanshrink	Tanshrink	Tanshrink	dSiLU

Table 1 shows the maximum and minimum values observed for each of the datasets. Overall, Dual Line prefers the range as 1.4E−03 to 0.363 and DP ReLU prefers 1E−03 to 0.251. Higher learning rates are observed in LENET4 compared to WideResNet and LENET5.

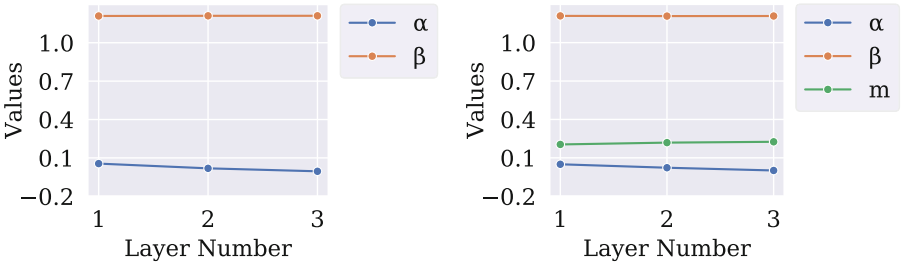
8.2 Parameter Value Analysis

The value of the learn-able parameters used in the activation function decides the response region, thereby the characteristics of the activation function. The neighboring blocks have an impact on the activation function as it receives input from them. The activation function requirement may vary based on the position of activation function within a block of a network, which can be analyzed by checking the parameter distribution of the activation function. As LENET5 and LENET4 models don't have repeating blocks, no clear patterns were perceived as shown in Fig. 5 and Fig. 6.



(a) DP ReLU activation function (b) Dual Line activation function

Fig. 5. Value of parameters w.r.t position of the activation function from top to bottom of LENET5 model

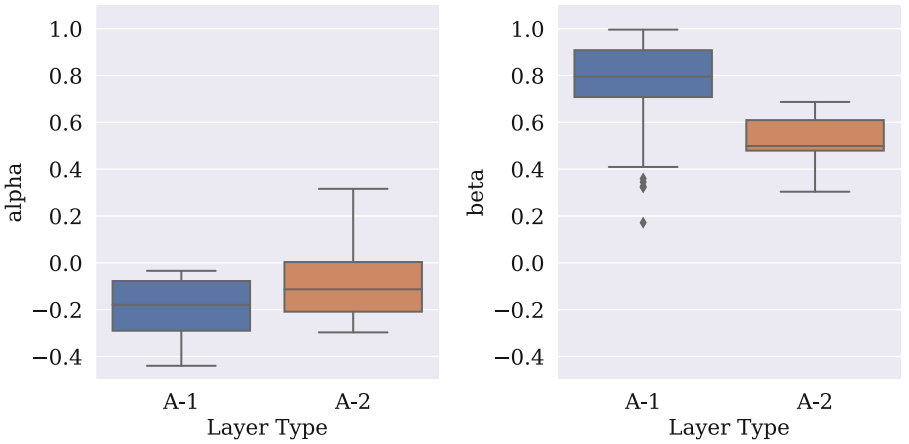


(a) DP ReLU activation function (b) Dual Line activation function

Fig. 6. Value of parameters w.r.t position of the activation function from top to bottom of LENET4 model

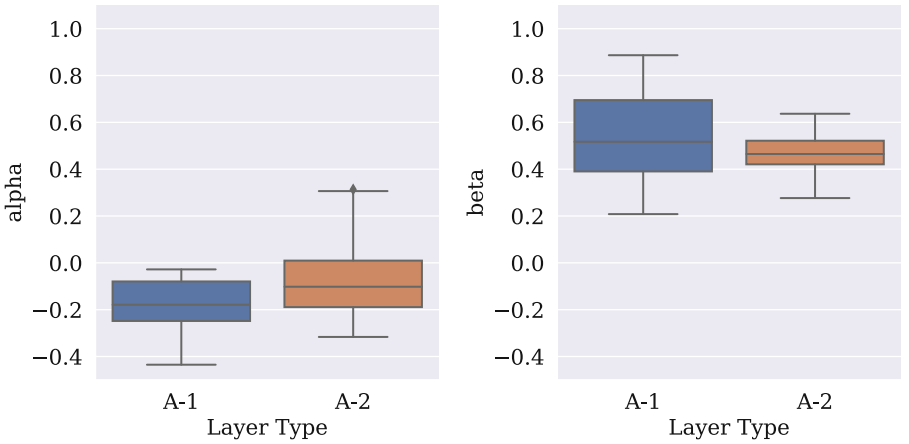
In WideResNet, activation function occurs twice in each of the nine repeating blocks of the network. The distribution of parameter values w.r.t each of these blocks is analyzed to view the relationship existing between activation functions occurring within a same block, as shown in Fig. 9a and Fig. 9b. The value of β

of the 1st activation function within a block is larger than the 2nd activation function in the blocks (B1–B8). The first block (B0) which is close to the input, does not exhibit this pattern.



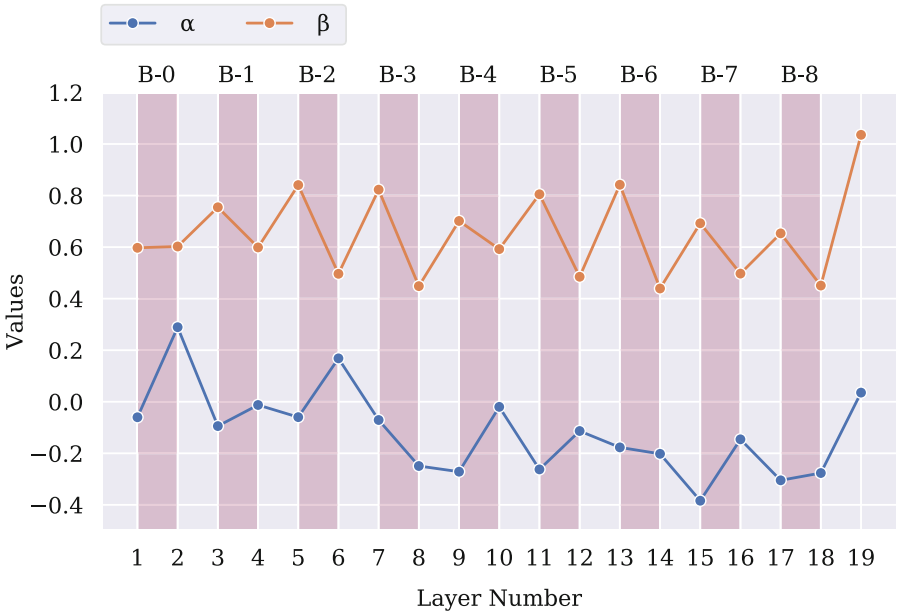
(a) Distribution of α parameter (b) Distribution of β parameter

Fig. 7. Box plot of parameters of DP ReLU activation within each block. A-1 and A-2 represent the first and second activation function within each block of WideResNet trained on CIFAR10

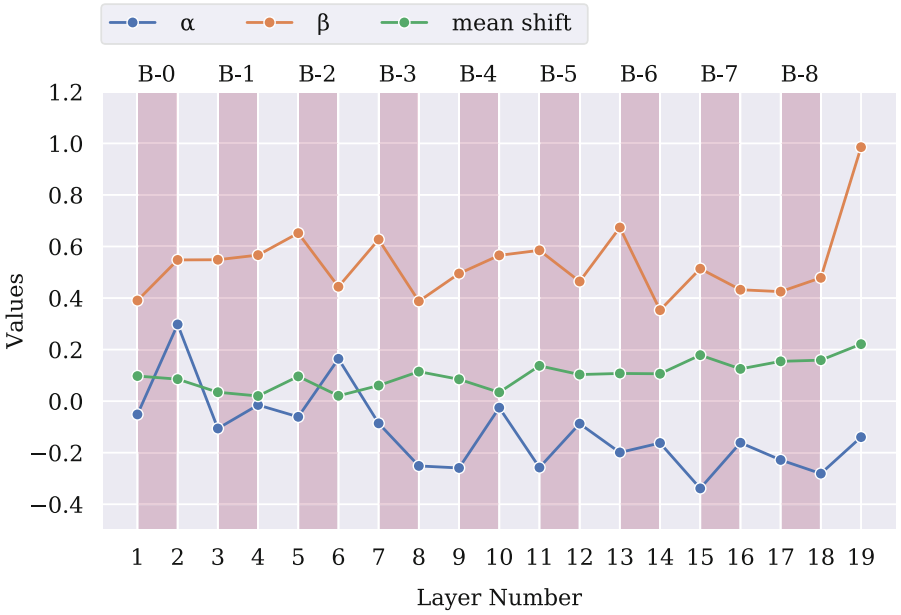


(a) Distribution of α parameter (b) Distribution of β parameter

Fig. 8. Box plot of parameters of Dual Line activation within each block. A-1 and A-2 represent the first and second activation function within blocks of WideResNet trained on CIFAR10



(a) Value of α and β parameters of DP ReLU trained on CIFAR10



(b) Value of α, β and mean shift parameters of Dual Line trained on CIFAR10

Fig. 9. Distribution of parameter values for proposed activation functions across blocks of WideResNet. X - axes represent the position of the activation function from top to bottom of the network. *Red vertical blocks* (B-0 to B-8) represent each of the blocks

Box plot analysis was done to understand the distribution of parameter values. The box plot of α value indicates the marginal difference in distribution within a block as shown in Fig. 7a.

The distribution of β value differs a lot within a block as shown in Fig. 7b. The outliers present in the box plot of β values are due to values from block (B0), which is close to input. This indicates that the activation function requirements differ within a block of a network.

From Fig. 9b, it can be observed that the α parameter does not exhibit any significant pattern, the mean shift parameter is nearly constant and β values differ marginally for Dual Line. The marginal difference in β distribution can be seen clearly in Fig. 8b. The block-level parameter distribution pattern indicates the final distribution of the parameters and helps us to understand the requirements of an activation function at the block level. The proposed activation function can adapt as per the requirement posed by its neighboring layers due to the inclusion of learn-able parameters.

8.3 Performance on fast.ai Leaderboards

The proposed activation function Dual Line broke 3 out of 4 fast.ai leaderboards for the image classification task, performing better than Mish activation function, which was the previous best performer. It was able to achieve more than 2% improvement in accuracy in two leaderboards. Parameter distribution pattern within blocks was observed in XResNet-50 models trained for the above 4 leaderboards. Models with Dual Line activation function took nearly 1.3 times more time for training compared to models with Mish. The major limitation for the current activation function is the time taken during the training.

In our current work, the concept of the learn-able slope parameter for the positive axis and mean shift parameter are analyzed and their performance benefit is shown. Our future work deals with reducing the computation time taken during training.

9 Conclusion

The novel concept of adding a learn-able slope and mean shift parameter is introduced in this paper. Overall, our experiments indicate the performance benefit of the proposed concept. The concept can be added to other activation functions with ease for performance boost. As the paper captures the activation function requirement at the block level, the proposed concept can be used as a supporting guideline for developing new activation functions for computer vision.

A Appendix

Table A shows the Name of activation functions and their expansion.

Activation Function	Expansion
ARiA2	Adaptive Richards curve weighted Activation
Atan	Arc-Tangent
BReLU	Bipolar Rectified Linear Unit
CELU	Continuously Differentiable Exponential Linear Unit
CReLU	Concatenated Rectified Linear Unit
DPReLU	Dual Parametric Rectified Linear Unit
dSiLU	Derivative of Sigmoid-Weighted Linear Unit
ELiSH	Exponential Linear Squashing Activation
ELU	Exponential Linear Unit
FTSwish	Flatten-T Swish
GELU	Gaussian Error Linear Unit
GReLU	General Rectified Linear Unit
ISRLU	Inverse Square Root Linear Unit
ISRU	Inverse Square Root Unit
LeakyReLU	Leaky Rectified Linear Unit
LiSHT	Linearly Scaled Hyperbolic Tangent
PELU	Parametric Exponential Linear Unit
PoLU	Power Linear Unit
PReLU	Parametric Rectified Linear Unit
PReLUC	Parametric Rectified Linear Units Channel-wise
ReLU	Rectified Linear Unit
RReLU	Randomized Rectified Linear Unit
SELU	Scaled Exponential Linear Unit
SiLU	Sigmoid-Weighted Linear Unit
SQNL	Square Non-Linearity Activation
SReLU	S-shaped Rectified Linear Unit
TReLU	True Rectified Linear Unit

References

1. Ichi Amari, S.: Natural gradient works efficiently in learning. Neural Comput. **10**, 251–276 (1998)
2. Carlile, B., Delamarter, G., Kinney, P., Marti, A, Whitney, B.: Improving deep learning by inverse square root linear units (ISRLUs). arXiv e-prints arXiv:1710.09967, October 2017

3. CIFAR-10 and CIFAR-100 dataset. https://www.cs.toronto.edu/~kriz/cifar.html
4. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). arXiv e-prints arXiv:1511.07289, November 2015
5. Fashion MNIST. https://www.kaggle.com/zalando-research/fashionmnist
6. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. arXiv e-prints arXiv:1502.01852, February 2015
7. Hendrycks, D., Gimpel, K.: Gaussian Error Linear Units (GELUs). arXiv e-prints arXiv:1606.08415, June 2016
8. ILSVRC2012. http://www.image-net.org/challenges/LSVRC/2012/
9. Labeled Faces in the Wild Home. http://vis-www.cs.umass.edu/lfw/
10. lessw2020: Trelu (2019). https://github.com/lessw2020/TRelu
11. Maas, A.L.: Rectifier nonlinearities improve neural network acoustic models. In: JMLR (2013)
12. Mixed Precision Training. https://docs.nvidia.com/deeplearning/sdk/mixed-precision-training/index.html#mptrain
13. THE MNIST DATABASE. http://yann.lecun.com/exdb/mnist/
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf
15. National Data Science Bowl. https://www.kaggle.com/c/datasciencebowl/data
16. Nwankpa, C.E., Ijomah, W., Gachagan, A., Marshall, S.: Activation functions: comparison of trends in practice and research for deep learning. arXiv e-prints arXiv:1811.03378, November 2018
17. PyTorch. https://pytorch.org/
18. Qiu, S., Xu, X., Cai, B.: FReLU: flexible rectified linear units for improving convolutional neural networks. arXiv e-prints arXiv:1706.08098, June 2017
19. Smith, L.N.: Cyclical learning rates for training neural networks. arXiv e-prints arXiv:1506.01186, June 2015
20. Dawnbench. https://dawn.cs.stanford.edu/benchmark/#cifar10
21. The Street View House Numbers (SVHN) Dataset. http://ufldl.stanford.edu/housenumbers/
22. Tiny ImageNet Visual Recognition Challenge. https://tiny-imagenet.herokuapp.com/
23. Trottier, L., Giguère, P., Chaibdraa, B.: Parametric exponential linear unit for deep convolutional neural networks. arXiv e-prints arXiv:1605.09332, May 2016
24. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: beyond empirical risk minimization. arXiv e-prints arXiv:1710.09412, October 2017

Drone-Based Cattle Detection Using Deep Neural Networks

R. Y. Aburasain[1(✉)], E. A. Edirisinghe[1], and Ali Albatay[2]

[1] Department of Computer Science, Loughborough University, Loughborough, UK
r.aburasain@lboro.ac.uk
[2] Zayed University, Dubai, United Arab Emirates

Abstract. Cattle form an important source of farming in many countries. In literature, several attempts have been conducted to detect farm animals for different applications and purposes. However, these approaches have been based on detecting animals from images captured from ground level and most approaches use traditional machine learning approaches for their automated detection. In this modern era, Drones facilitate accessing images in challenging environments and scanning large-scale areas with minimum time, which enables many new applications to be established. Considering the fact that drones typically are flown at high altitude to facilitate coverage of large areas within a short time, the captured object size tend to be small and hence this significantly challenges the possible use of traditional machine learning algorithms for object detection. This research proposes a novel methodology to detect cattle in farms established in desert areas using Deep Neural Networks. We propose to detect animals based on a 'group-of-animals' concept and associated features in which different group sizes and animal density distribution are used. Two state-of-the-art Convolutional Neural Network (CNN) architectures, SSD-500 and YOLO V-3, are effectively configured, trained and used for the purpose and their performance efficiencies are compared. The results demonstrate the capability of the two generated CNN models to detect groups-of-animals in which the highest accuracy recorded was when using SSD-500 giving a F-score of 0.93, accuracy of 0.89 and mAP rate of 84.7.

Keywords: Drones · Object detection · Convolution Neural Networks · Unmanned aerial vehicles

1 Introduction and Literature Review

The surveillance of large areas/spaces such as farms and deserts has become easier through the emerging technology of Unmanned Aircraft Vehicles (UAVs). A drone is a specific, low-cost and simple UAV that significantly facilitates monitoring difficult-to-access environments [1]. Drones used in agriculture and livestock monitoring applications have a significant potential to support the respective market sectors according to 'Fox-business' that has reported the current and future estimates of the US investigations on using drones for digitalizing farms, managing live-stock and improving general facilities [2]. These technologies could also be useful in countries with large

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 598–611, 2021.
https://doi.org/10.1007/978-3-030-55180-3_44

areas of desert such as the countries in the Gulf region, e.g. UAE, Saudi-Arabia etc. Such countries could use drone technologies supported by the latest development of AI technology to develop agriculture and effectively exploit their natural resources in support of agriculture, which this research aims to contribute to.

For the proposed research to be conducted, a Drone-based dataset was captured in a desert area with the aim of using the captured data for automating the detection of animals using Deep Neural Network models. To ensure compliance with aviation regulations, safety and enable the coverage of a large area within the shortest time, drones are usually flown at a high altitude, which presents the challenge that animals or any other object to be very small, thus making it very challenging to detect any features that will enable the detection or recognition of the objects. As it is illustrated in Fig. 1, the object features cannot be clearly deciphered using the human eye. Since the aim is to localise the animals but at the same time not necessarily counting them, we propose a localisation model that uses features of a group-of-animals in detecting them. This method is adopted due to the fact that the features of a single animal are not captured in an adequate manner to distinguish and localise each single animal. Hence, combining the features from multiple animals within a group can have a significant impact on the detection of a group-of-animals. However, there is still a challenge: animals are moving objects and this makes their distribution and numbers of animals within a specified area vary with time. Our aim is to investigate how to best use the popular CNN architectures, SSD and YOLO, to establish learnt models that are capable of detecting groups of any type of animal, any size, at density of distribution of animals within the space.

Fig. 1. Samples of farm animals in the given dataset. (A)–(C) show group of goats at different sizes and distributions, (D) shows group of camels.

In general for object detection tasks, DNNs have been proven to outperform conventional machine learning methods [3]. Several application areas have attracted the attention of both practitioners and academics, who have effectively deployed DNNs in video surveillance, autonomous driving, rescue and relief operations, the use of robots in industry, face and pedestrian detection, understanding UAVs images, recognising brands and text digitalisation etc. Motivated by the success of DNNs in such application areas, we propose to use DNNs for drone-image analysis in support of livestock monitoring in farms. It is pointed out that there has not been any previous attempts in analysing the

drone footage in detecting groups of animals captured at low resolutions and there are no pre-trained DNNs therefore developed for this purpose. Thus the work presented in this paper is both novel and contributes significantly to the advancement of the subject area.

The history shows that the first attempt in applying deep learning in object detection was conducted when a *R-CNN* was developed which combined the selective search for generating 2000 *region proposals* with convolutional neural network for classification. The proposed system comprised of three main steps; starting by selective search for generating 2000 region proposals and using 2000 CNNs to generate the region feature maps and ending with the application of a Support Vector Machine (SVM) that classified the regions.

However, the main drawback of the above system is the time consumed which significantly improved in *Fast R-CNN* [4] by running one CNN in pre-processed images and replacing the SVM by extending the CNN with a softmax layer for classification. Rather than using the selective search, the third version which called *Faster R-CNN* [5] uses region proposal network (RPN) to generate the proposed regions before using either ResNet or inception for classification. The series *R-CNN*, *Fast R-CNN* and *Faster R-CNN* are the most popular in this community. Even the detection speed has significantly improved in the *Faster R-CNN*, it still separates the procedure into two steps using two networks; region proposal network (RPN) followed by Fast-R-CNN.

On the other hand, rather than separating the region selection and classification into two steps, Single-Shot Detector (SSD) conducted by [6] and You Only Looks Once (YOLO) developed by [7] perform the detection totally different by using a single convolutional neural network for detection and classification from the raw pixel values directly. As this is the focus of this research, the networks architectures are explained in detail in the methodology section.

Three different versions of YOLO has been published in which the first version has 24 conv layers followed by two fully connected layers. In YOLO-V2 [8], the authors improve the localization accuracy by adding the batch normalization to the conv layers, increase the image resolutions and using the anchor boxes to predict the bounding boxes rather than using the fully connected layers. However, SSD performs better than YOLO V-2 as the predicted boxes for each location is higher than in the first two versions of YOLO. To compete, YOLO's authors develop YOLO V-3 [9], which outperform SSD in several datasets including the benchmark COCO dataset. In YOLO-V3, the architecture is changed by increasing the conv layers to 106, building residual blocks and skipping the connection to improve the detection at different scale. Also, they change the squared errors in the loss function to cross-entropy terms and replacing the softmax layer with logistic regression which predict the label given a threshold value. Following this historical development in object detection using deep learning, SSD and YOLO-v3 have been selected and compared to detect cattle in the given dataset.

Despite the advancement of deep learning in general object detection tasks, few attempts are published for Drone-based cattle detection. Most of the previous activities employ standard machine learning algorithms including SVM and HOG, such as in [10] and [11]. Usually, camera trap images or regular images are used as in [12, 13] and [14].

Considering the fact that Drones has several advantages compared to either camera traps or satellite images and deep learning outperformed other machine learning techniques, applying them to cattle detection is still under investigation, we have noted that there is a scarcity information on this area. To the best of our knowledge, the few attempts presented below have been published under the criteria mentioned.

In [2], authors make use of images captured in a Namibian wildlife reserve park. The pre-trained model in AlexNet architecture in combined with their few detection layers is used for the detection. The results show that a substantial improvement is achieved when a comparison is made to the standard Fast R-CNN model.

In [15], authors also attempt to make use of a particular drone referred to as Multirotor for purposes of detecting cattle. These authors make use of custom network structure and intensive data augmentation that proliferate the sample to 3600 images from 300. The authors mention that the framework they employ is able to deliver a significant result for only one class detection since the network they use is simpler with fewer convolutional layers. In a similar way, [16] focus area is the counting and detecting of cattle where the route of the drone is known beforehand so that the overlapping between frames (images) can be estimated. A record of the specific flying route is kept so that the detected objects are not counted twice on different images. As the environment is totally different from ours, YOLO V-2 gives a considerable result in their experiment while its performance is quite low in our dataset.

Notwithstanding the broad array of comparisons between results obtained from different deep learning object detection architectures in varying applications such as in [17, 15] and [18], there still remains a dearth of explanations as to the reason why particular net performs better than others with regards to accuracy. Another aspect that will have a substantial influence that has not been discussed is the training strategies. Hence, this research seeks to contribute in the following ways.

- Improve the detection of cattle in drone-based images where the animals' size are relatively small by extracting group-of animals' features as opposed to single animals.
- Apply, evaluate, and compare the performance of SSD-500 and YOLO V-3 as two prominent architectures in single-shot object detection for the given dataset.
- Present a discussion of the two architectures hyper-parameters that may have an impact on performance.

The rest of this paper comprises of four sections. The configuration of the research dataset is presented in Sect. 2 while the proposed research methodology is introduced in Sect. 3. In Sect. 4, the experiment result and discussion is shown followed by the conclusion and proposed future work in Sect. 5.

2 Research Dataset

The full research dataset comprised of 221 large dimensional Drone-based images captured in a desert area. The pixel resolution of captured images is 5 cm $\times$ 5 cm per pixel on the ground. The dimensions of a captured drone image is, 5000 $\times$ 3000 (see Fig. 2).

For the purpose of conducting this research a sample of the above data was used. 300 images were collected from the full research dataset, dimensioned at 608 × 608. The captured images include only the top view of animals and the animal features are not clear, no facial or body details are visible that will allow the recognition or even the detection of the animals. This reflects the difficulty in recognizing each single animal, where the bounding box for each animal do not exceed 12 × 12 pixels in the collected 608 × 608 images. There are not enough discriminative features for each single animal to enable their use in training millions of parameters in a DNN to create a model to detect a single animal.

However, a group-of-animals show rich features that can easily and effectively be used to determine where animals are located in a desert/farm area. Therefore, we have proposed to localize animals based on the group-of-animals features. A Dataset-1 300 images is collected from the full research dataset with 1760 bounding boxes around areas comprises of either groups-of-goats or groups-of-camels at different densities of distribution and bounding box sizes. The animals in some groups are quiet close to each other while in some cases they are not.

Fig. 2. Sample of the desert animal farm dataset, dimensions 5000 * 3000 pixels.

3 Methodology

The focus of this section is to present the details of configuration and training of the chosen two competing DNN architectures, SSD-500 and YOLO-v3, in order to create models that will be capable of detecting groups-of-animals. These architectures were chosen from a possible set of other available DNN architectures following a careful process of considering the specific dataset, the object detection requirements and the

architecture of the networks. Considering the reality that the dataset has restricted volume of data for training, with small size of objects and low resolution details providing low quality objects, these architectures take into consideration the detections at varying scales at different stages. Since the primary aim of this application is to find out where the animals are located without counting them, we have come up with detecting animals in groups by extracting features within the group. This section will begin by providing details of network configurations of YOLO v-3 and SSD-500, DNN architectures.

3.1 SSD vs. YOLO Architectures

The authors in [6] introduced Single-Shot Detection (SSD) for the task of object detection based on using VGG-16 architecture developed by [19] for feature extraction followed by few additional layers for detection of the objects based on the extracted features. In practice SSD has two versions: SSD-300 and SSD-500. The main difference between these two is the input image's minimum dimension. When a comparison is made between SSD-500 and SSD-300, it can be noted that the accuracy of the former is better than that of the latter. This is because starting from the top layers, a larger sized receptive field is used in SSD-500. Therefore, only the latter is considered for the purposes of this study.

The structure of the SSD network begins with VGG-16 convolutional neural networks used in regular object recognition tasks and by replacing the final two connected layers with auxiliary convolutional layers. In SSD architecture, after extracting the features using VGG-16, the fourth convolutional layer contains a feature map of size $38 \times 38 \times 512$ and SSD predicts four bounding boxes for each location with depth being disregarded. The result of this is (4 predictions) multiplied by the (boundary box dimensions) multiplied by the (confidence score for all classes). It is important to note that since this is an early phase of prediction at the feature map size of 38×38, a resolution that is far from the original image resolution of 512×512, the expectation is that the results obtained from detecting each individual animal at a dimension of 12×12 is not going to be adequate for detection.

With regards to the YOLO architecture, there are three versions in which train at different scales have been considered in YOLO V-3. This network contains of 106 convolution layers based on darknet-53. It concatenates channels using shortcut connections and residual blocks. While SSD predicts four bounding boxes for every location in the feature map, YOLO V-3 predicts three for the three defined scales. What this implies is that while SSD can determine four bounding boxes for every location in each feature map, YOLO concatenates the channels at three stages and predicting three at three different scales.

So, SSD employs less conventional layers when compared to YOLO V-3 which has 106 convolutional layers. However, they employ a specific strategy, as has been discussed above, to localise an object. Hence, it is important in this particular study area to understand whether the number of convolutional layers or increasing the prediction and estimation at each single location is significant in this research dataset. This understanding calls for a comparison and experimenting of the performance, a task that will be done in the experimentation section.

Both YOLO-v3 and SSD employ cross-entropy for purposes of calculating loss and giving the confidence loss (L_{conf}). The computation of localisation loss (L_{loc}) is achieved using L2-Norm by compared to the ground truth, as is illustrated in Eq. 1, where N denotes the matched boxes.

$$L(x, c, l, g) = \frac{1}{N}(Lconf\,(x, c) + \alpha\, Lloc(x, l, g))$$

Both YOLO V-3 and SSD make use of the Non-Maximum Suppression (NMS) method in the final phase for purposes of eliminating the larger quantity of boxes predicted at each of the feature map's locations. This means boxes with a particular IoU threshold and confidence loss are eliminated so that predictions that are nosier can be removed.

3.2 Proposed Methodology

Based on the discussion above in regard to YOLO V-3 and SSD-500, theses two architectures have been applied in the collected dataset which comprises of 300 images with dimension 608 × 608 as follows:

1. Defining bounding boxes using LabelImg open source in which each box comprises of group-of-animals. Variant sizes are used to insure there is sufficient training samples for such case. These boxes have been saved into two formats; TXT and XML for YOLO V-3 and SSD-500 training respectively.
2. Using 90% of the given dataset for training and 10% for testing and uses the exact training and testing data to train SSD and YOLO V-3.
3. Creating the LMDB for SSD training.
4. Adjusting SSD and YOLO network parameters for single class detection and tuning the batch size for optimization.
5. Monitoring the Average loss for early stopping if necessary.
6. Calculating the mAP, precision, recall, F-score and overall accuracy

4 Results and Discussion

This experiment has been conducted using Nvidia GTX - 1070 graphics card and Ubuntu 18.04 operating system. Caffe framework has been used for training SSD and Darknet is used for YOLO V-3. Starting by collecting 300 images from the given desert dataset, 1760 bounding boxes has been labelled as in Fig. 3. Based on the fact that training a very deep net require a sufficient number of samples, the number of samples is increased by defining more overlapped bounding boxes with different centre point when the animals appeared not too closed to each other.

The same annotations have been saved into two formats, TXT and LMDB, for YOLO and SSD training respectively. This has been divided into 90% and 10% for training and testing. Given the 30 testing images, there are 161 bounding boxes for testing.

Fig. 3. Sample of Group Animals annotations using LabelImg.

In Table 1, the main parameters and specific batch size, subdivision and accumulate batch size is presented. It is shown in this table that YOLO V-3 uses a slight larger image which reflect a higher receptive field used from the top layer. While SSD uses a slightly smaller image compared to YOLO V-3, it still predicts four bounding boxes for each location compared to YOLO V-3 which predict only 3. On the other hand, the maximum batch size we able to uses for each network is 12 and 4 in YOLO V-3 and SSD-500 respectively. The higher batch size allows to take more samples to calculate the gradient and we have noticed how the result is dropped out when the batch size $= 1$. Despite this fact, this parameter is also depending on the dataset and working environment. It is well known in the community to tune it between 12, 24 and 32. However, larger batch size requires a powerful memory or otherwise, the out of memory will be the case for terminating the process.

Table 1. SSD-512 and YOLO V-3 parameters used in this experiment.

	YOLO V-3	SSD-500
Dimension	608 * 608	512 * 512
Conv layers	106	29
Learning rate	0.001	0.0004
Learning decay	0.0005	0.0005
Activation function	Relu	Leaky
Batch size	12	2
Subdivisions	4	_
Accumulate batch size	_	2

The average loss has been monitored in each architecture to stop the training once the model starts to add noise to its parameters. It is noted that this technique which called Early stopping is crucial in the given dataset. Therefore, it has been applied to select the best model with lowest loss. While the loss average in YOLO V-3 is 2.71 after 1000 iterations, it was 17.2 in SSD-500. The optimal model with lower loss in YOLO-V3 is 0.22 after 60,000 iterations compared to 0.36 obtained after 160K iterations. In contrast, the average loss using SSD-500 is gradually decreased and the lowest average loss is 1.83 after 160K iterations.

To evaluate the learnt models, the precision, recall and F1-score are calculated using Eqs. 1, 2, 3 and 4. The interpretation of these terms is shown in Table 2.

$$Presion = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1\ score = 2.\frac{Presion.Recall}{Presion + Recall} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

Table 2. The interpretation of performance evaluation terms.

True Positive (TP)	The number of correctly detected G-Animals
True Negative (TN)	Correctly not detecting and localize not G-Animals
False Positive (FP)	The number of false detected G-animals
False Negative (FN)	The number of missed detections
Precision	The ratio of correctly detected Group-Animals to the total detection; True and False
Recall	The ratio of correctly detected Group-Animals to baseline or ground truth Group-animals in the given dataset

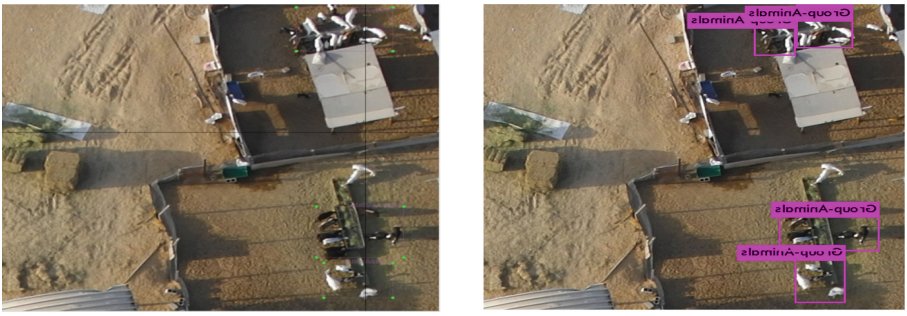
However, unlike conventional object detection problems where the object location is precisely localized in the image, the group-of-animals at different distributions and numbers is vary. As the result is basically evaluated on unseen images, the detection using the learnt model can be different from the annotations. It has been noted the efficiency of the leant model for detecting group-of-animals different from the annotations, especially

when the animals are sparsely distributed. Figure 4 shows sample of the pre-defined annotation and detection result of the learnt YOLO V-3 model. Therefore, evaluating the detection for each test image has been conducted to insure the *TP*, *TN* and *FP* are accurately calculated.

The result of training YOLO V-3 and SSD-500 in the given dataset is presented in Table 3 and Fig. 7. It is shown that the True Positive (TP) is improved in SSD compared to YOLO V-3. On the other hand, the precision is 1 in YOLO V-3 where the FP is 0 which reflects there is no false detection in the test set. While SSD-500 gives higher *TP* rate compared to YOLO V-3, there is a very small FP in the test set as shown in Fig. 5. On the other hand, the average confidence rate is 5% better in YOLO V-3 because the final average loss was lower in YOLO compared to SSD.

Table 3. The result of training SSD and YOLO in the given dataset in the term of Precision, recall and F1-score.

	TP	FN	Precisions	Recall	Avg. confidence	F1 score
YOLO V-3	131	30	1	0.81	89.68	0.89
SSD-500	144	17	0.98	0.89	84.7	**0.932**



(A) (B)

Fig. 4. Difference of Data annotations and detections results. Both cases are corrected in estimating and detecting Group-animals.

Despite the higher TP rate in SSD-500, the precision in YOLO V-3 is better. In Fig. 6, Sample of the FP in SSD-500 is presented.

(A) (B)

Fig. 5. Sample of the missed True Positive (TP) detection in YOLO V-3 (B) compared to SSD-500 (A).

Fig. 6. Sample of the FP sample using the learnt SSD-500 model. The false detection (FP) is shown in the top right corner.

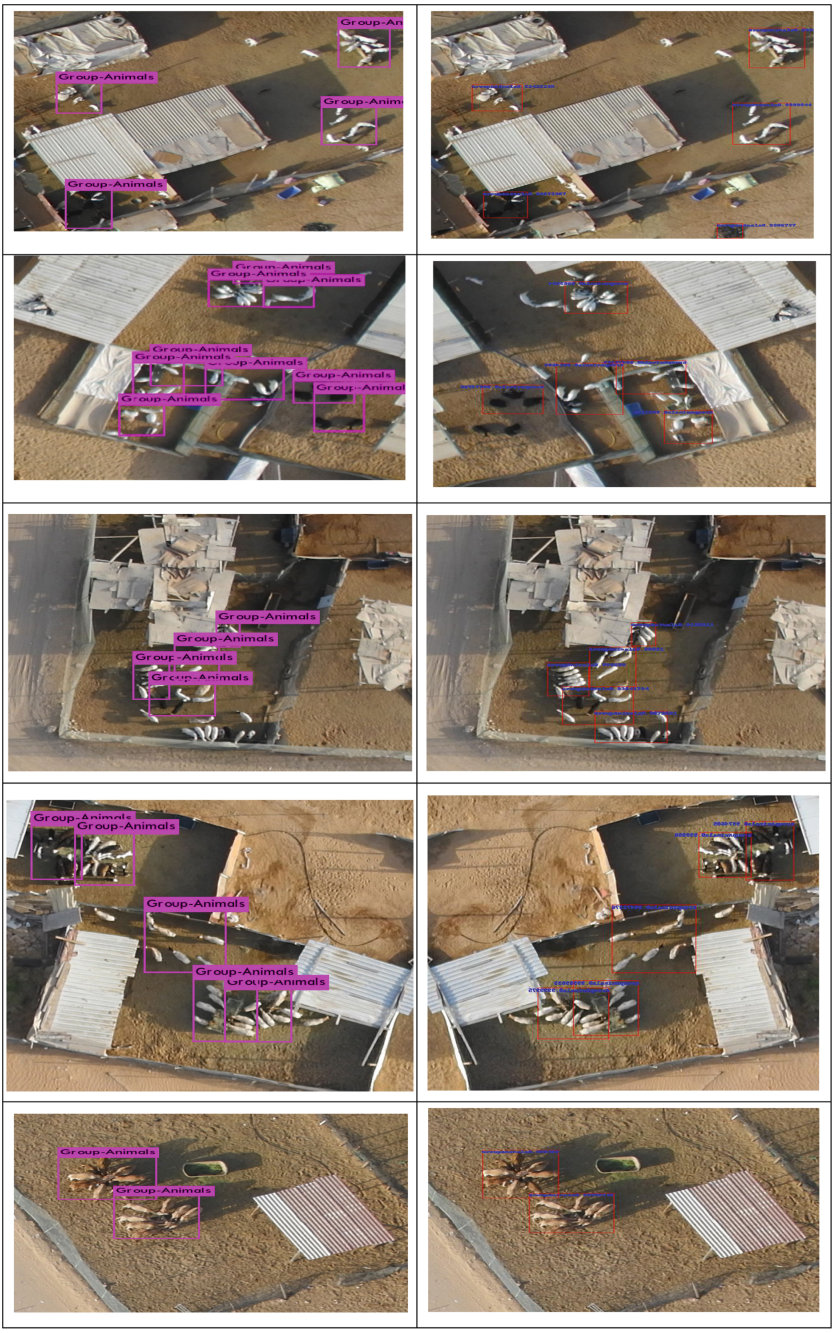


Fig. 7. Final Result of testing YOLO V-3 and SSD-500 learnt models in left and right columns respectively. This reflect the models efficiently on detecting and localizing any size, type and distribution of group-of animals using the proposed methodology.

5 Conclusion and Future Work

In this research, a cattle detection based on the state-of arts DNN architectures in object detection, SSD-500 and YOLO V-3, is proposed. Cattle detection in drones-based can be performed by training group-of-animals features rather than training each single animal separately. Labelling more boxes for each cattle area with different size improve the result of detected and localized animals. The SSD-500 outperformed YOLO V-3 in this studied dataset because it predicts more boxes for each location in the features map. The result indicates the possibility of using the proposed strategy for localizing animals in Drone-based even the images resolution is low with no clear features appeared. As an extension for this work, the authors tend to detect multiple classes with the aim to investigate the applicability of the CNNs in analysing drone footage in such areas.

References

1. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. Networks **72**(4), 411–458 (2018). https://doi.org/10.1002/net.21818
2. Kellenberger, B., Volpi, M., Tuia, D.: Fast animal detection in UAV images using convolutional neural networks. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 866–869 (2017)
3. Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E.: Deep learning for computer vision: a brief review. Comput. Intell. Neurosci. **2018**, 1–13 (2018)
4. Girshick, R.: Fast R-CNN. In: 2015 Presented at the Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448. Available: http://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html. Accessed 26 Apr 2019
5. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28, pp. 91–99. Curran Associates, Inc. (2015)
6. Liu, W., et al.: SSD: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37 (2016)
7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
8. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger, December 2016. https://arxiv.org/abs/1612.08242v1. Accessed 13 May 2019
9. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. ArXiv Prepr. ArXiv180402767 (2018)
10. Sharma, S.U., Shah, D.J.: A practical animal detection and collision avoidance system using computer vision technique. IEEE Access **5**, 347–358 (2016)
11. Nasirahmadi, A., Edwards, S.A., Sturm, B.: Implementation of machine vision for detecting behaviour of cattle and pigs. Livest. Sci. **202**, 25–38 (2017)
12. Yousif, H., Yuan, J., Kays, R., He, Z.: Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), May 2017, pp. 1–4. https://doi.org/10.1109/iscas.2017.8050762

13. Gomez Villa, A., Salazar, A., Vargas, F.: Towards automatic wild animal monitoring: identification of animal species in camera-trap images using very deep convolutional neural networks. Ecol. Inform. **41**, 24–32 (2017). https://doi.org/10.1016/j.ecoinf.2017.07.004
14. Norouzzadeh, M.S., et al.: Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proc. Natl. Acad. Sci. **115**(25), E5716–E5725 (2018). https://doi.org/10.1073/pnas.1719367115
15. Rivas, A., Chamoso, P., González-Briones, A., Corchado, J.M.: Detection of cattle using drones and convolutional neural networks. Sensors **18**(7), 2048 (2018). https://doi.org/10.3390/s18072048
16. Shao, W., Kawakami, R., Yoshihashi, R., You, S., Kawase, H., Naemura, T.: Cattle detection and counting in UAV images based on convolutional neural networks. Int. J. Remote Sens. **41**(1), 31–52 (2020). https://doi.org/10.1080/01431161.2019.1624858
17. Xia, M., Li, W., Fu, H., Yu, L., Dong, R., Zheng, J.: Fast and robust detection of oil palm trees using high-resolution remote sensing images. In: Automatic Target Recognition XXIX, May 2019, vol. 10988, p. 109880C. https://doi.org/10.1117/12.2518352
18. Hollings, T., Burgman, M., van Andel, M., Gilbert, M., Robinson, T., Robinson, A.: How do you find the green sheep? A critical review of the use of remotely sensed imagery to detect and count animals. Methods Ecol. Evol. **9**(4), 881–892 (2018)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ArXiv Prepr. ArXiv14091556 (2014)

Anomaly Detection Using Bidirectional LSTM

Sarah Aljbali(✉) and Kaushik Roy

Department of Computer Science, North Carolina A&T University, Greensboro, NC 27411, USA
saramub.L@gmail.com, kroy@ncat.edu

Abstract. This paper presents an anomaly detection approach based on deep learning techniques. A bidirectional long-short-term memory (Bi-LSTM) was applied on the UNSW-NB15 dataset to detect the anomalies. UNSW-NB15 represents raw network packets that contains both the normal activities and anomalies. The data was preprocessed through data normalization and reshaping, and then fed into the Bidirectional LSTM model for anomaly detection. The performance of the BLSTM was measured based on the accuracy, precision, F-Score, and recall. The Bi-LSTM model generated high detection results compared to other machine learning and deep learning models.

Keywords: Intrusion detection · LSTM · Bidirectional LSTM · Deep learning

1 Introduction

Over the last decade, Machine learning (ML) approaches have been used to detect intrusions. However, the needs of significant human effort for feature engineering impeded the real-world applications of ML in the area of intrusion detection [1]. Intrusion detection system (IDS) is classified as Active IDS or Passive IDS based on their responsive nature. An active IDS works by automatically blocking the malware attacks, without the need of human intervention; on the other hand, passive IDS alert the users after monitoring the network traffic [2]. IDS can also be classified into Signature-Based IDS and Anomaly based IDS. The IDS accesses a database of known signatures and vulnerabilities in the signature based approach. To detect and prevent future attacks, attack signature that contains details of each intrusion attack is used. The need of the database updates frequently for the new and unseen is the major disadvantage of this approach. On the other hand, the Anomaly-based IDS detect new intrusions by learning from the baseline patterns. Attacks alarms are triggered when there is any deviation from the existing baseline pattern [2]. IDS can be classified based on the place in which it is mounted. It can be network intrusion detection system when an IDS is located on the network section, whereas, IDS considered host-based when it is deployed in workstations. There are many disadvantages in using host-based IDS mentioned in [2]. Network Intrusion Detection Systems (NIDSs) are necessary tools to detect different security attacks inside an organization's network [3].

Deep learning, or deep structural learning, is a form of machine learning that is broader in its structure, complexity, and representations of how the data learn [4].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 612–619, 2021.
https://doi.org/10.1007/978-3-030-55180-3_45

It consists of layers of deep neural networks through which features are learned hierarchically and automatically [5]. To find the patterns in the network, deep learning algorithms need an enormous data amount compared to traditional machine learning algorithms [2]. Deep Learning is useful to detect anomalies in IoT network with diverse and multi-modal data [12].

Traditional machine learning algorithms fail to deliver results for IoT devices, which are connected for long time-durations, for extended period of time [2]. The capability to learn from the previous time-steps and the less dependency on human intervention is what recurrent neural networks (RNN) [2] offer. At each time-step in RNN, each node provide an output that will be an input to the same node in the hidden layer. Information that is useful is kept in the memory used later in the future time steps for learning purposes [2].

Deep learning algorithms transform long short-term memory networks (LSTMs) from recurrent neural networks (RNNs) to an effective model in learning patterns in long serious [6]. LSTM models is more effective than RNN models, when data has a long reliance on time [7]. This long reliance can be found in IoT applications [8]. LSTM networks can be used in learning features and patterns to classify network data as 'benign' or 'attack'. Since it operates on raw data, LSTM as deep learning algorithm minimize the difficulty of feature engineering over traditional ML. Adversaries will not be able to be adapted to feature learning algorithms to promote their intrusion techniques, therefore the LSTM network is flexible when compared to adversaries. LSTM is effective on unstructured training data that can be used for Internet of Things (loT), while most of the DL algorithms work on numeric datasets [5].

A further development of LSTM is the Bidirectional long short term memory (Bi-LSTM) [9]. Bi-LSTM can access both the previous and following contexts by merging the front and the back hidden layers. LSTM only exploits the historical context compared to Bi-LSTM. Therefore, Bi-LSTM can be better than LSTM in solving the sequential modelling task [9]. The purpose of this paper is to provide performance evaluation of Bidirectional LSTM on UNSW-NB15 dataset [9].

2 Related Work

Many researchers described the use of ML and DL in anomaly detection that reported high detection rates. In [10], the LSTM was applied along with Gradient Descent Optimization on the KDD 99 dataset, and the model achieved an accuracy of 97.54% and recall of 98.95% [10]. In [11] authors applied LSTM on the KDD 99 data set and obtained sufficient result. The Long Short Term Memory (LSTM) structure has been applied to a Recurrent Neural Network (RNN), it is used to train the IDS model with KDD Cup 99 dataset. By performing the test, it was found that the deep learning method LSTM is efficient for IDS by achieving the highest accuracy of other classifiers 96.93% [6]. Most of the state-of-the-art use KDD 99 [12] and NSLKDD [13] datasets to test the performances of the IDSs. However, KDD99 dataset suffers from data duplication. There is a duplication in about 78% and 75% of the train and test set records in KDD dataset [12]. As a result, the learning algorithms will be biased towards in the train and test sets. The NSL-KDD dataset [13] removed the redundant data, however, fails to represent the actual networks and associated attack scenarios.

In this research, we used UNSW-NB15 dataset [14] to evaluate the performance of our IDS. The UNSW-NB15 dataset consists of real modern normal behaviors and current synthesized attack activities. Training and testing sets of this dataset have same probability distribution. The UNSW-NB15 contains the network packets which involve a set of features from the payload and header of packets. In [15], authors applied different ML algorithms on UNSW-NB15 dataset, and decision tree obtained a highest accuracy of 85.65%.In [12], authors also applied some traditional ML algorithms, including support vector machines (SVMs), Naïve Bayes and Random Forest on UNSW-NB15 dataset. The random forest archived a highest accuracy of 97.49%. This paper will investigate the Bi-LSTM performance, which is extension of LSTM, using UNSW-NB15 dataset.

3 Methodologies Used

This section descries the methods and dataset used in this paper for network anomaly detection.

3.1 LSTM

LSTM extended from RNNs [8]. Although the advantages of the RNNs of having the ability of using information context between the sequences of input and output,the range of contextual information it may store is limited [16]. Therefore, information propagation will be affected and result in reducing impact of hidden layers [17]. To overcome this limitation of being dependent in a long-term and having a vanished gradient, the LSTM was proposed [18]. Every cell in LSTM considered as memory block [13] the memory block is consisted of: (a) forget gate ft, (b) input gate It, and (c) output gate, The forget gate remove the memories that are irrelevant according to the cell state, the updated information will be controlled by the input gate in the LSTM cell, and the output gate controls how the output is filtered. The logistic sigmoid and network output functions are denoted by σ and tanh, respectively. Furthermore, the LSTM output layer will be connected to the LSTM cell output [16].

3.2 Bidirectional LSTM

The Bi-LSTM neural network is an extension of Bidirectional Recurrent Neural Network (BRNN). Bi-LSTM also is an extension form of the one-way LSTM. For prediction purpose, it utilizes additional information. The same output layer is connected by forward and a backward layer as shown in Fig. 1. There is no connection between forward and backward cell states, future information will be included when there is no introduction of delays [16].

3.3 UNSW-NB15 Dataset

For creation of UNSW-NB 15 dataset, IXIA PerfectStorm tool used to collect a mixture of modern normal and current intrusion activities of network traffic [19]. The dataset has two categories: normal and attack. The attack category involves nine attacks [19].

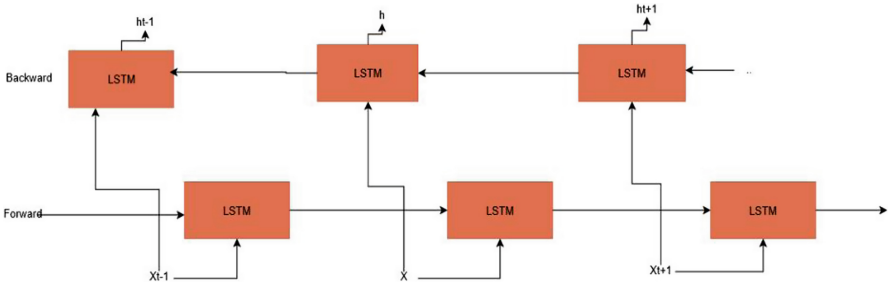


Fig. 1. Bi-LSTM structure [16]

Part of the dataset is segregated into training set and testing set. The training set contains 175,341 records, and the testing set contains 82,332 records. The training and testing sets contain all types of records that is categorized as normal and attacks [19].

The nine attack types involved in the dataset shown in Table 1: Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode, and Worm.

Table 1. b UNSW-NB15 dataset distribution [19]

Category	Training set	Testing set
Normal	56,000	37,000
Analysis	2000	677
Backdoor	1746	583
DoS	12,264	4089
Exploits	33,393	11,132
Fuzzers	18,184	6062
Generic	40000	18871
Reconnaissance	10491	3496
Shellcode	1133	378
Worms	130	40
Total records	175,341	82332

3.4 Data Processing

To process the data to fit the LSTM and BiLSTM model, the following steps are followed:

1. Use dummy encoding to convert unique strings in [‘proto’, ‘service’, ‘state’] to numerical features. Also transform the categorical values in [‘attack_cat’] column in training and testing sets to numerical values.

2. Labels and categories ['attack_cat', 'label'] moved to the end columns. Also, features in the two subsections of datasets need to be in the same order.
3. Dropna() function used to drop 'nan' values in training and testing sets to avoid training errors.
4. Normalizing all numerical features: with mean 0 and standard deviation 1.
5. Input reshape [samples, time steps, features]: LSTM and Bi-LSTM input must be as the following (num sample, time steps, num features) format, therefore reshape() function is used on the NumPy array.

4 Results and Discussions

4.1 Evaluation Measures

The measures used to study the Bi-LSTM performance are discussed in this section. Table 2 describes the classification of data. In [20], the categorization of measures used to analyze the data is identified.

Table 2. Data classification [20]

		Predicted	
		Attack	Normal
Actual	Attack	TP	FN
	Normal	FP	TN

Accuracy can be identified as the rate of data categorized accurately, for which actual attack data are classified as attacks and normal data as normal [20]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TP} + \text{FP} + \text{FN}} \quad [20]$$

Precision measures the correct amount of positive predictions by dividing the number of true positives by the number of predictive positives. Recall measures the percentage of actual positive cases by dividing the true positive by the number of actual positives. The output value of precision and recall is between 0 and 1. It is always desired in intrusion detection systems to have high recall and precision values. Low false positives and false negatives are recommendable assessment criterion in an attack detection system because they indicate high relevancy [5].

ROC the receiver operating characteristic (ROC) curve, defined in [21], is widely used and an effective method of evaluating the performance of the model.

Sensitivity and specificity as defined in [21] are used to evaluate the AUC. AUC is a measure of the overall performance of model and is identified as the average value of sensitivity for all possible values of specificity [22]. It can take on any number between 0 and 1. It is recommended for AUC to be closer to 1. This provides a positive indication of the model performance. When AUC value of a model is 1, it means it is perfectly accurate [21].

The loss function described in [23]. The lower the **loss,** the better a model. The main objective in a learning model, with respect to the model's parameters, is to decrease the loss function's value [23].

4.2 Performance Evaluation

To analyze performance of variant classification algorithms on UNSW_NB15 dataset for the binary classification: Random Forest, Support Vector Machine, LSTM and Bi-LSTM are used to train models through the training set (using 10-layer cross-validation). The models then applied to the testing set. The results are mentioned in Table 3. Compared with other classifiers, the accuracy of Bi-LSTM classification show the highest rate of 99.66. Bi-LSTM also shows high value for both precision and recall as shown in Table 4.

Bi-LSTM ROC curve in Fig. 2 shows AUC value of 1 for 10 folds.It means the performance of Bi-LSTM is highly accurate. Figure 3 shows two line plots for the loss function over epochs for the LSTM validation (green) and Bi-LSTM validation (red) datasets. Figure 3 implies that Bi-LSTM behave well because the loss value decreases for several iterations.

Table 3. Accuracy rates of different used approaches.

Approach	Accuracy
SVM	97
LSTM	99.66
Bi-LSTM	99.70
RF	98

Table 4. Bi-LSTM classification report.

Class	Class	F1-score	Precision
Precision	1	1	1
Attack	1	1	1

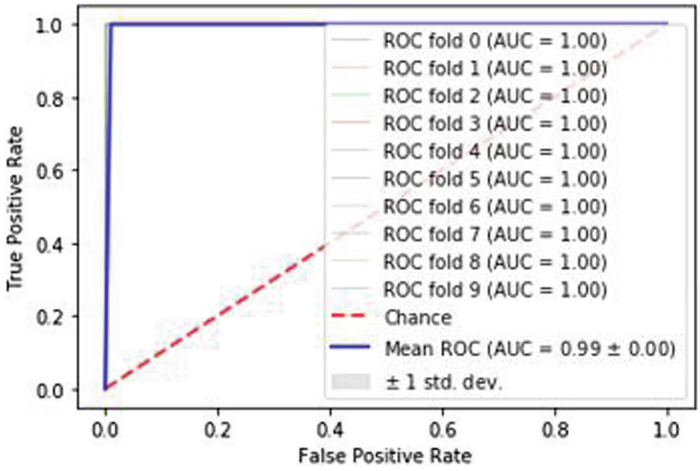


Fig. 2. Bi-LSTM ROC curve

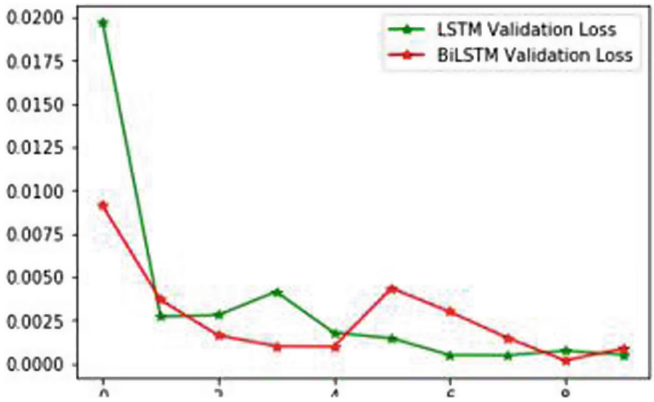


Fig. 3. LSTM vs Bi-LSTM validation loss

5 Conclusion

In this paper the deep learning is utilized to develop efficient and flexible intrusion detection system. The long short term memory model has been described and its extension Bidirectional LSTM, both show strong intrusion detections modeling ability, the models show also high accuracy in binary classification. Compared with traditional classification approaches, such as SVM and random forest, the result of BI-LSTM achieves better accuracy rate and detection rate with a low false positive rate under the task of binary classification on the UNSW15 dataset. The Bi-LSTM model can advance the accuracy of intrusion detection efficiently. In the forthcoming papers, the performance of LSTM and Bidirectional LSTM classification using other dataset in the network-based intrusion detection systems will be evaluated.

Acknowledgments. This research is based upon the work supported by the CISCO systems, Inc.

References

1. Yin, C., et al.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **5**, 21954–21961 (2017)
2. Manoj, P.: Deep learning approach for intrusion detection system (IDS) in the internet of things (IOT) network using gated recurrent neural networks (GRU) (2011)
3. Quamar, N., Weiqing, S., Ahmad, J., Mansoor, A.: A deep learning approach for network intrusion detection system
4. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
5. Diro, A., Chilamkurti, N.: Leveraging LSTM networks for attack detection in fog-to-things communications. IEEE Commun. Mag. **56**(9), 124–130 (2018)
6. Malhotra, P., et al.: Long short term memory networks for anomaly detection in time series. In: Proceedings of European Symposium on Artificial Neural Networks Computational Intelligence and Machine Learning, pp. 22–24 (2015)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555v1 [cs.NE] (2014)
8. Mohammad, M., Alfuqaha, A., Sorour, S.: Deep learning for IoT big data and streaming analytics: a survey (2017)
9. Liu, G., Guo, J.: Bidirectional LSTM with attention mechanism and convolutional layer for text classification (2019)
10. Kim, J., Kim, H: An effective intrusion detection classifier using long short-term memory with gradient descent optimization. In: International Conference on Platform Technology and Service (PlatCon), pp. 1–6. IEEE (2017)
11. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
12. Belouch, M., Hadaj, S., IdHammad, M.: Performance evaluation of intrusion detection based on machine learning using apache spark. Proc. Comput. Sci. **127**(C), 1–6 (2018)
13. Sak, H., Senior, A.W., Beaufays, F.: Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Proceedingsof Annual Conference on International Speech Communication Association (INTERSPEECH), pp. 338–342, September 2014
14. Moustafa, N., Slay, J:A hybrid feature selection for network intrusion detection systems: central points (2017)
15. Moustafa, N., Slay, J: The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set (2016)
16. Li, Z., Batta, P., Trajkovic, L.: Comparison of machine learning algorithms for detection of network intrusions (2018)
17. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3), 229–256 (1992)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
19. Moustafa, N., Slay, J.: The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems (2015)
20. Kim, J., Shin, N., Jo, S.Y., Kim, S.H.: Method of intrusion detection using deep neural network. In: IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, pp. 313–316 (2017)
21. Park, S.H., Goo, J.M., Jo, C.H.: Receiver operating characteristic (ROC) curve: practical review for radiologists (2004)
22. Zhou, X.H., Obuchowski, N.A., McClish, D.K.: Statistical Methods in Diagnostic Medicine, 1st edn, pp. 15–164. Wiley, New York (2002)
23. Godoy, D.: Understanding binary cross-entropy/log loss: a visual explanation (2018)

Deep Neural Networks: Incremental Learning

Rama Murthy Garimella[1(✉)], Maha Lakshmi Bairaju[2(✉)], G. C. Jyothi Prasanna[2(✉)], Vidya Sree Vankam[2(✉)], and Manasa Jagannadan[2(✉)]

[1] Mahindra Ecole Centrale, Hyderbad, India
rama.murthy@mechyd.ac.in
[2] IIIT RGUKT, RK Valley, Kadapa, Andhra Pradesh, India
lakshmiiiit49@gmail.com, prasannajyothi805@gmail.com, vankamvidyasree@gmail.com, manasajagannadan@gmail.com

Abstract. In this research paper, the problem of Incremental Learning is addressed. Based on the idea of extracting features incrementally using Auto-Encoders, CNNs, Deep Learning architectures are proposed. We implemented proposed architectures on Raw dataset (containing collection of main classes and dummy classes) and compared the results with CIFAR10 dataset. Experimental investigations are reported.

Keywords: Incremental learning · Auto-encoders · Convolutional Neural Networks · Classification

1 Introduction

The research area of Computational Neuro-Science encompasses Artificial Neural Networks (ANNs). The first stage of progress on designing and implementing ANNs culminated in the back propagation algorithm utilized in the Multi Layer Percepton (MLP) implementation. The next stage of progress on ANN's was initiated with the deep learning paradigm. Specifically, Convolutional Neural Networks showed excellent progress in achieving better than human accuracy in many real world classification problems. But, CNNs are far away from being able to achieve functions performed by the human brain. For instance, the human brain acquires knowledge incrementally in classification, association and many other tasks. Thus, the human brain is endowed with "incremental learning" ability. This research paper is an effort in achieving incremental learning based on Deep Neural Networks.

This research paper is organized as follows.

In Sect. 2, known related Literature is briefly reviewed. In Sect. 3, CNN architectures to learn one object at a time are discussed. In Sect. 4, auto-encoder based IL architectures are discussed. Also CNN based IL architectures are discussed.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 620–629, 2021.
https://doi.org/10.1007/978-3-030-55180-3_46

2 Review of Related Research Literature

Human brain has the ability to acquire knowledge (classification, association, memory etc tasks) from natural physical reality INCREMENTALLY. Researchers are thus motivated to study models of INCREMENTAL LEARNING (IL) using Artificial Neural Networks (ANNs). Adaptive Resonance Theory (ART) is dedicated to enable INCREMENTAL LEARNING [3]. The first author attempted the problem using ensemble classifier models [4–6]. There were some successful results that were reported. This research paper is a culmination of such efforts. Some researchers reported Incremental Learning (IL) using Convolutional Neural Networks [1,2].

3 Novel Convolutional Neural Network (CNN) Architecture: Learning One Object at a Time

The innovation in the CNN architecture for Incremental Learning (IL) is summarized below.

- The input images have only one object such as a CAT. It also has a dummy object so that there are two classes. The Convolutional & Pooling layers are trained using such input achieving good accuracy.
- A separate ANN is trained (i.e. The Convolutional & Pooling layers are trained) using images containing a single different object such as a DOG and a dummy object.
- The trained architectures based on CNNs are fed to fully connected layers (Dense layers) in a parallel architecture. Such a novel architecture is fed with input images containing both a CAT and DOG separately.

The testing accuracy is determined with such ANN.

Remark: The goal is to enable the ANN to incrementally learn new objects while remembering the existing knowledge.

Here we introduced different Deep Leraning architectures built with Convolutional Neural Networks and Autoencoders. These architectures perform well in the classification by providing good training and validation accuracies.

4 Incremental Learning Architectures

In this section, we describe various deep learning architectures we have experimented with for incremental learning.

4.1 Auto-Encoder Based Architecture for IL

Architecture-1

In this architecture-1, we consider classification problem with finitely many classes.

Step1: Auto-Encoders (particularly convolutional) are trained (i.e. Encoder-decoder combination is utilized to extract features through nonlinear dimensionality reduction) individually for each class.

Step2: The encoder parts (for each class) are stacked in parallel and fed to fully connected layers. Such a stacked architecture is then fed with objects belonging to various classes and the validation accuracy is determined.

The architectures which we used in the model are depicted in the below diagrams (Fig. 1 and 2). In this Convolutional Autoencoder we used 6 Convolutional layers, 3 pooling layers and 2 sampling (upsampling) layers. We used 3×3 kernels in the Convolutional layers and 2×2 kernels in the pooling(maxpool) layers. Here in this architecture of Convolutional Autoencoder we give images of shape (256, 256) as input. We train the encoder and decoder part of Convolutional Autoencoder with the dataset of images belonging to one class like cats. Repeat the same procedure for the other class like dogs. Take the encoder parts from both the models and flatten them.

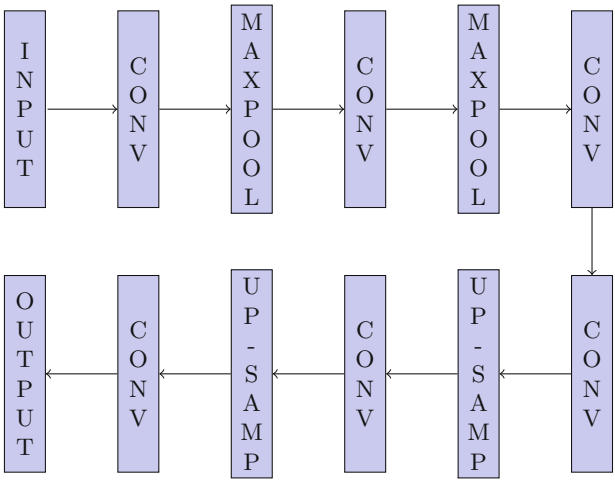


Fig. 1. Block diagram of model-1 for Architecture-1.

Put the trained autoencoders in parallel and feed them to fully connected layers. The first fully connected layer comprises of 256 neurons with Relu as the activation function the output of which is connected to second fully connected layer with 128 neurons by using the activation as Relu. Then a dropout of 0.2 is used. The final fully connected layer is included with 2 neurons as the output with Softmax as the activation function. In the final merged model we used loss function as the binary cross entropy with optimizer Adam (Fig. 3).

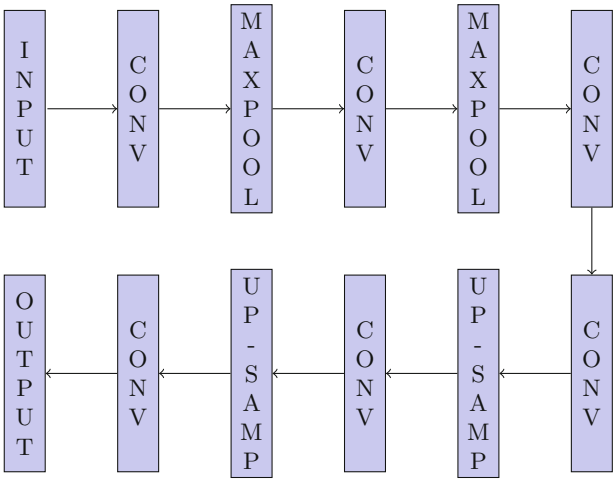


Fig. 2. Block diagram of model-2 for Architecture-1.

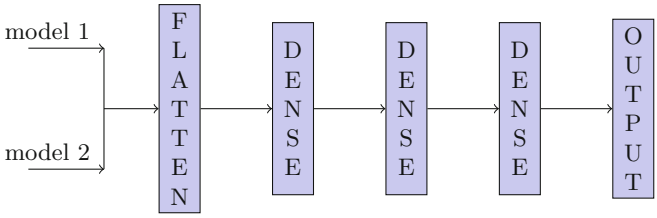


Fig. 3. Final block diagram of Architecture-1.

Here are the details of CNN architectures.

4.2 CNN Based Architecture for IL

Architecture-2

The effective idea is to train CNN's incrementally and feed them to fully connected layers for incremental classification.

The Convolution Neural Network is having five Blocks and three Dense layers including output layer. Each Block contains one Convolution layer, one Pooling layer and one Batch Normalization layer. So, this Convolution Neural Network is having five Convolution layers, five Pooling Layers, five Batch Normalization layers and three Dense layers including output layer (Fig. 4).

Fig. 4. Block diagram of block.

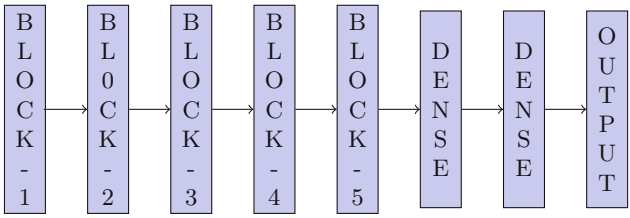


Fig. 5. Block diagram of model-1 for Architecture-2.

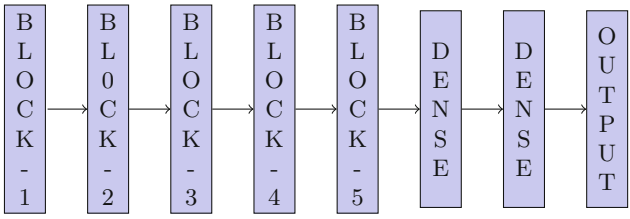


Fig. 6. Block diagram of model-2 for Architecture-2.

In this architecture, we have used 3 × 3 as kernel size with different depth for convolution layers and Max-Pooling with 2 × 2 as kernel size. The first Dense layer is having 256 neurons, second Dense layer is having 128 neurons and Output layer contains N neurons with Softmax as activation function (where N represents number of classes).

In Architecture-2, for two-class classification, we had taken two models (Fig. 5 and 6). We extracted features in CNN for each class separately. Then we merged these models and their output is given as input to three Dense layers including Output layer. The first Dense layer is having 256 neurons, second Dense layer is having 128 neurons and Output layer contains N neurons with Softmax as activation function (where N represents number of classes) (Fig. 7).

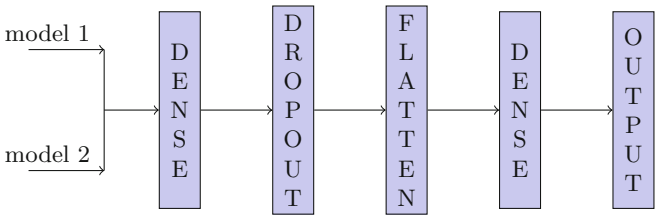


Fig. 7. Final block diagram of Architecture-2.

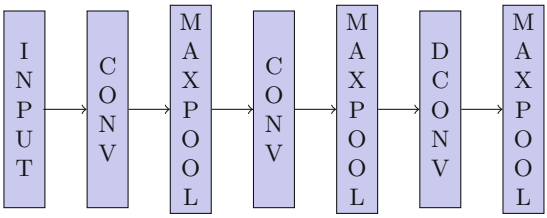


Fig. 8. Block diagram of model-1 for Architecture-3

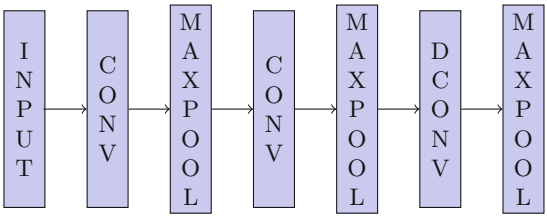


Fig. 9. Block diagram of model-2 for Architecture-3

Architecture-3

We had developed another CNN based architecture with three Convolutional layers, three Max Pooling layers. We used 3×3 kernels in the Convolutional layer & 2×2 kernels in the Maxpooling layers (Fig. 8 and 9).

We train this CNN model with the raw dataset of images having two classes (main class and dummy class) with input image shape of (64, 64) for classifying main class. Repeat this procedure to classify another main class with another raw dataset of images having another main class and dummy class. Put these trained CNNs in parallel and feed them to fully connected layers having 128 neurons. And Output layer contains N neurons with Softmax as activation function (where N represents number of classes) (Fig. 10).

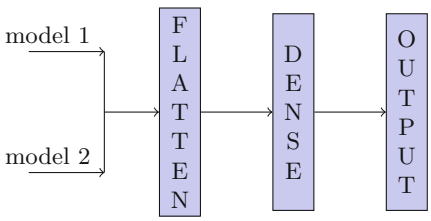


Fig. 10. Final Block diagram of Architecture-3

4.3 Multi-class IL (Fig. 11)

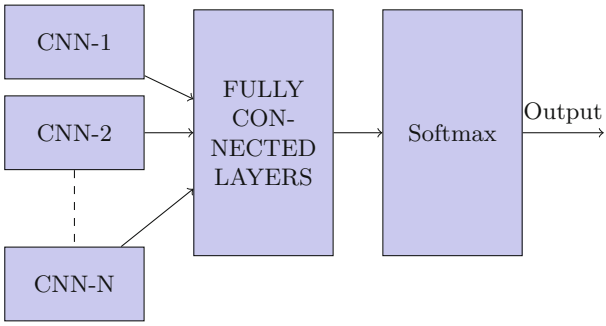


Fig. 11. Block diagram of MULTI-CLASS IL

- Train a CNN with multiple classes i.e, training phase and validation phase with good accuracy is completed (e.g. 95% accuracy).
  e.g. Multiple classes correspond to different animals : Horses, cats, etc.
- Extract convolutional & pooling layer outputs with freezed weights i.e. Trained CNN-1.
- Train other CNN on non living objects i.e, Extract convolutional and pooling layers with freezed weights.
- Put CNN-1, CNN-2 in parallel and feed to fully connected layers.

Need for IL

- Number of classes is unknown ahead of time.
- The trained network need not be retrained after new objects are presented to network.

Architecture-4
Train the CNN for 4 classes and give it to fully connected layers (Fig. 12).

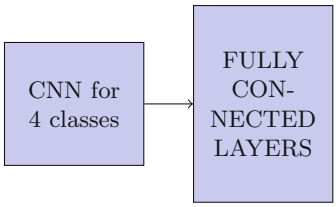


Fig. 12. Block diagram of 4-class CNN for Architecture-4

Train the CNN for 5 classes and give it to fully connected layers (Fig. 13).

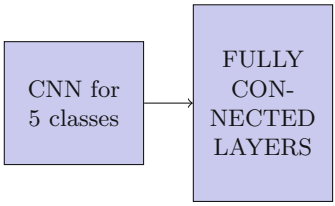


Fig. 13. Block diagram of 5-class CNN for Architecture-4

Architecture-5: Architecture-5
Train 4 classes with CNN (mentioned in Architecture-4) and train the 5th class with Auto-Encoder (AE) (Fig. 14). And give the output of the two corresponding models to the fully connected layers.

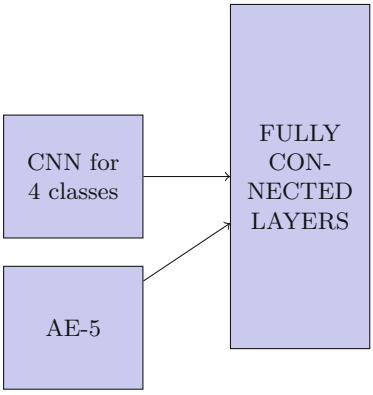


Fig. 14. Block diagram of Architecture-5

Architecture-6 (Fig. 15)

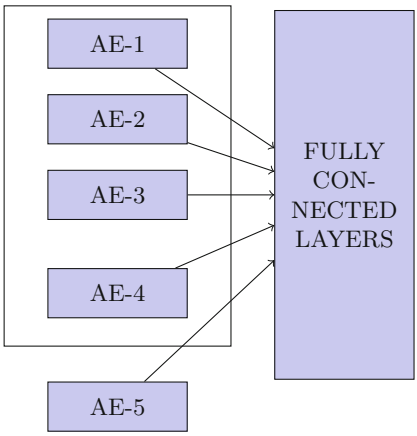


Fig. 15. Block diagram of Architecture-6

Architecture-7
Train classes 1, 2, 3, 4 with Auto-Encoders AE-1, AE-2, AE-3, AE-4, respectively. Train the 5th class with separate Auto-Encoder. And give the output of the two corresponding models to the fully connected layers.

Train two classes (main class and dummy class) with CNN-1 followed by Auto-Encoder-1. Again train another two classes (main class and dummy class) with CNN-2 followed by Auto-Encoder-2. Put these two trained models in parallel and feed them to fully connected layers (Fig. 16).

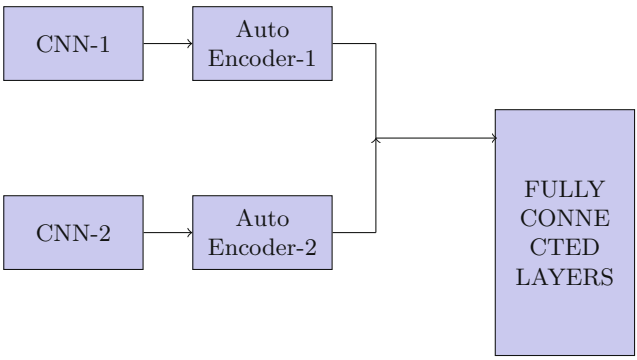


Fig. 16. Block diagram of Architecture-7

5 Experimental Results

We had trained our Architectures on raw dataset (contains main class and dummy class) and CIFAR 10 dataset. The accuracies of the given architectures are mentioned in the below table (Table 1).

Table 1. Experimental results of all Architectures

Architecture	Raw dataset		CIFAR10	
	Train acc	Val acc	Train acc	Val acc
Architecture-1	99%	77%	92%	91.5%
Architecture-2	93%	71%	93.5%	88.52%
Architecture-3	88%	57%	90%	90%
Architecture-4	97%	78%	93.5%	88.52%
Architecture-5	76.56%	69%	93.5%	88.52%
Architecture-6	94%	81%	84.69%	80.36%
Architecture-7	78.57%	78.06%	94.27%	90.26%

6 Conclusions

In this research paper using various Deep Learning architectures, Incremental Learning is demonstrated. We observed that some architectures are giving better accuracy on training with Raw dataset than CIFAR10 dataset. We are actively investigating novel ANN architectures for improving the classification accuracy.

References

1. Roy, D., Panda , P., Roy, K.: Tree-CNN : a hierarchial deep convolutional neural network for incremental learning. Accepted Neural Netw. (2019)
2. Castro, F.M., Marin-Jimenez, M.J., Guil, N., Schmid, C., Alahari , K.: End-to-end incremental learning. In: ECCV (2018)
3. Asfour, Y.R., Carpenter, G.A., Grossberg, S., Lesher, G.W.: Fusion ARTMAP: an adaptive fuzzy network for multi-channel classification. In: Proceedings of the Third International Conference on Industrial Fuzzy Control and Intelligent Systems (IFIS) (1993)
4. Bairaju, S.R., Ari, S., Garimella, R.M.: Facial emotion detection using deep autoencoders. In: Proceedings of International Conference on Recent Innovations in Electrical, Electronics and Communications Engineering (ICRIEECE), July 2018
5. Bairaju, S.R., Ari, S., Garimella, R.M.: Emotion detection using visual information with deep auto-encoders. In: IEEE 5th International Conference for Convergence in Technology (Scopus Indexed) (2019)
6. Basha, K.I., Garimella, R.M.: Emotion classification: novel deep learning architectures. In: Proceedings of IEEE International Conference on Advanced Computing & Communication Systems (ICACCS), 15th–16th March 2019. IEEE Explore (2019)

SNAD Arabic Dataset for Deep Learning

Deem AlSaleh(✉), Mashael Bin AlAmir(✉), and Souad Larabi-Marie-Sainte(✉)

College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia
deemsaleh@gmail.com, mshael.alameer@gmail.com, slarabi@psu.edu.sa

Abstract. Natural language processing (NLP) captured the attention of researchers for the last years. NLP is applied in various applications and several disciplines. Arabic is a language that also benefited from NLP. However, only few Arabic datasets are available for researchers. For that, applying the Arabic NLP is limited in these datasets. Hence, this paper introduces a new dataset, SNAD. SNAD is collected to fill the gap in Arabic datasets, especially for classification using deep learning. The dataset has more than 45,000 records. Each record consists of the news title, news details, in addition to the news class. The dataset has six different classes. Moreover, cleaning and preprocessing are applied to the raw data to make it more efficient for classification purpose. Finally, the dataset is validated using the Convolutional Neural Networks and the result is efficient. The dataset is freely available online.

Keywords: Dataset · Arabic text · Deep learning · Classification · Natural language processing

1 Introduction

Natural language processing (NLP) is one of the cores of artificial intelligence. It is formulated to process the human language. Processing the human language arises in applications like text classification, text summarization, speech recognition, and others. Having data is the first and most significant phase for creating such applications. Applications can have different languages based on the human language selected. The Arabic language is one of the semantic languages and the official language of 22 countries [1].

Arabic NLP researches are limited as there is a lack of Arabic datasets. Hence, Natural Language Processing for the Arabic language is bounded in a specified number of datasets. Moreover, the count of instances for these datasets is not large hindering the use of Deep Learning models which have proven to provide high classification results. Due to the lack of Arabic datasets, this paper introduces a new Arabic dataset called Saudi Newspapers Articles (SNAD) Arabic Dataset. The dataset is collected from the Saudi Press Agency (SPA) and AlRiyadh Newspaper websites. It is available for the scientific community.

This research presents a brief background of the Arabic language. The following section covers the current available Arabic text datasets. Then the new

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 630–640, 2021.
https://doi.org/10.1007/978-3-030-55180-3_47

dataset SNAD is presented along with its features, format, cleaning and preprocessing steps. At the end, SNAD is validated using the Convolution Neural Network classification technique.

2 Arabic Language

The Arabic language is spoken by more than 250 million people as it is one of the semantic languages and the language of the Islamic religion [2]. For Muslims, Arabic is essential since it is the language of Qur'an, the holy book for Muslims, which is read by none native Arabs as well. Additionally, Arabic is the native language of Middle Eastern and North African countries. It has several dialects due to the large geographic area for people who speak Arabic. Although various dialects exist, Arabic has one written script in which sentences are written from right to left. The Arabic language has 28 different letters that can be connected forming different words and sentences. Vowel, Nunation, and Shadda are the main properties of the Arabic language that can be used to provide different interpretations of sentences. Another characteristic of the Arabic language is the shape of letters which is based on their positions in the word and also the sentence. Several other characteristics of the Arabic language exist causing ambiguity and complexity and raising various challenges in processing the Arabic language [3].

3 Related Work

This section goes over the studies that developed Arabic datasets. Only six datasets have been found.

3.1 SANAD

Single-Label Arabic News Articles dataset (SANAD) is a recent dataset published in 2019 for the purpose of text classification. The dataset was collected from three newspaper websites, AlArabiya, Akhbarona, and AlanbaAlKhaleej. It has seven categories, which are culture, finance, medical, politics, religion, sports and technology. The dataset is organized into three folders, each represents the articles' source. Inside each folder, there are sub-folders for the different categories and each holds the different text files [4]. All articles are in their raw format as scraped from the webistes. The distribution of data is illustrated in Table 1. Though the number of articles in the dataset is large, it is not validated using machine learning models. Additionally, the dataset must be preprocessed so it can be used and the cleaning will be very time consuming due to the dataset size.

Table 1. SANAD dataset categories distribution

Category	Text files
Finance	45856
Sports	44935
Culture	18865
Technology	23110
Politics	24847
Medical	23162
Religion	14022
TOTAL	194797

3.2 Moroccan Dataset

The Moroccan newspaper dataset, proposed by [5] for the purpose of classification. It is a collection of 111,728 Moroccan articles extracted from three Arabic online newspapers: Assabah [9], Hespress [10], and Akhbarona [11]. The articles are distributed into five different categories including sports, politics, culture, economy, and others. The dataset is provided as a csv file with two columns, the text and the target class. The distribution of classes is illustrated in Table 2. After collecting the data, the authors applied some pre-processing to prepare the data for classification. The pre-processing steps included removing the digits, punctuation marks and stopwords, and also stemming to get the root words. Moreover, they used Convolutional Neural Networks (CNN) as a Deep Learning technique to classify the dataset, in addition to Support Vector Machine (SVM) and Linear Regression (LR) as Machine Learning Classifiers. The classification resulted an accuracy rate of 92.94%, 88.2%, and 86.3% for CNN, SVM, and LR respectively. However, the authors did not remove null values from their data which can affect the classification results.

Table 2. Moroccan newspapers articles dataset documents details

Resource	Sports	Politic	Culture	Economy	Diverse	Total
Assabah	34,244	2,381	5,635	2,620	9,253	54,133
Hespress	6,965	5,737	3,023	3,795	7,475	26,995
Akhbarona	5,313	12,387	5,080	7,820	0	30,600
Total	46,522	20,505	13,738	14,235	16,728	111,728

3.3 NADA Dataset

NADA dataset was developed by [6] for the purpose of classification. the dataset is collected from two Arabic datasets OSAC and Diab Dataset (DAA).

The DAA dataset contains nine categories, each category consists of 400 documents. In addition, OSAC is divided into six categories, each category consists of 500 to 3000 documents. The OSAC had raw Arabic data which required some pre-processing such as removal of digits, punctuation marks, none Arabic characters and Arabic stopwords. Also, light stemming was applied to get the root of the words. The final outcome of the dataset was published as CSV file. Moreover, it includes ten categories which are Arabic Literature, Economical Social Sciences, Political Social Sciences, Law, Sports, Art, Islamic Religion, Computer Science, Health, and Astronomy with a total number of 7310 text files. The distribution of text files is illustrated in Table 3. After finalizing the dataset, WEKA is used to measure the performance of applying classification to NADA. It was tested using SVM classifier which resulted an accuracy rate of 93.8792% in a duration of 24 min and 28 seconds. NADA is a well organized dataset which is pre-processed and ready for classification, however it only consists of 7,310 documents which will not be beneficial with Deep Learning methods.

Table 3. NADA dataset categories distribution

Category	Text files
Arabic literature	400
Economical social sciences	1307
Political social sciences	400
Law	1644
Sports	1416
Art	400
Islamic religion	515
Computer science	400
Health	428
Astronomy	400
Total	7,310

3.4 TALAA-ASC

The TALAA-ASC courpus [7] was created for the purpose of sentence compression for Arabic Natural Language Processing. The corpus includes five article categories, management and work, medicine, news and politics, sports, and technology. A total of 70 articles were collected from newspaper websites. The distribution of categories is illustrated in Table 4. The XML (eXtensible Markup Language) structure is used to represent the data as illustrated in Fig. 1. As observed in the figure, there are two versions of each article, the first is the original article while the second is the compressed version.

3.5 TALAA

TALAA is an Arabic dataset published in 2015 [7] for the purpose of classification. The dataset contains a total of 57,827 articles extracted from several public newspaper websites. These articles are enclosed under 8 topics, culture, economics, politics, religion, society, sports, world, and others. The distribution of article topics is illustrated in Table 5. After obtaining all the articles, the authors refined the dataset structure by segmenting the articles, applying feature selection techniques, then using the eXtensible Markup Language (XML) as a format. The TALAA dataset has a convenient number of instances, however, the topics World and Others are vague. For instance "a spanish football team became a champion in one of its games", what is the topic of this article? sports or world? maybe both!. Additionally, the datasaet is not available unless authorized.

Table 4. TALAA-ASC dataset categories distribution

Category	Text files
Management and work	10
Medicine	11
News and politics	14
Sports	10
Technology	25
Total	70

<?xml version="1.0" encoding="UTF-8"?>
<doc id="sport.12">
<original id="technology.12.0.S">
تفوقت شركة سامسونغ على شركة ءبل في مبيعات الهواتف الذكية، خلال الأشهر الثلاثة الأولى من العام، لتنهي بذلك شركة الإلكترونيات الكورية الجنوبية احتكار أنوكيا، أكبر مصنع للهواتف المحمولة في العالم من حيث حجم التصنيع.
</original>
<compressed id="technology.12.0.C">
تفوقت شركة سامسونغ على ءبل في مبيعات الهواتف الذكية، خلال الأشهر الثلاثة الأولى من العام، لتنهي بذلك احتكار نوكيا.
</ compressed >

Fig. 1. TALAA-ASC corpus structure sample

Table 5. TALAA dataset categories distribution

Category	Text files
Culture	5322
Economics	8768
Politics	9620
Religion	4526
Society	9744
Sports	9103
World	6344
Other	4400
Total	57827

3.6 MSA Dataset

The Modern Standard Arabic (MSA) dataset was built in 2014 [8] for the purpose of classification and clustering. The dataset articles were collected from different news sources based on the category of the article. Nine categories were created including art, economy, health, law, literature, politics, religion, sports, and technology. Each category has 300 articles resulting in a total of 2700 articles. Table 6 illustrates the distribution of the articles in the MSA corpus. Each article in the dataset is saved in a text file along with its category name. The authors provided five versions of the dataset. The first version provides the raw article script as is. The following provides a refined version with the Arabic script without common stop words and punctuation. The third and fourth versions provide the same articles but with different stemming techniques. Finally, the last version provides the articles after extracting their words' roots. The authors of [9] used the MSA dataset to classify Arabic text using an advanced version of Support Vector Machine (SVM). The obtained classification accuracy was 90.62%. Additionally, in [10], the authors used the same dataset for the purpose of comparing different feature selection techniques. Furthermore, the dataset was used in [11] for clustering Arabic text providing an F1-Measure and purity equal to 87.32% and 93.3% respectively. Since the number of instances in the MSA dataset is small, it was used for different purposes as classification, clustering, feature-selection and especially in proposing new machine learning models. However, the number of instances of MSA dataset is not sufficient when applying deep learning models.

3.7 Summary

The discussed related work presented several Arabic datasets. However, some of these [6,8] have a small number of instances which weakens the classification results. Such dataaets are not suitable for deep learning methods. Additionally, all of these contain raw instances requiring the user to process and clean the

dataset which is time and effort consuming. importantly, dealing with raw data consumes a large amount of memory which hinders the researchers from using it. In addition, not all the discussed datasets published their work to be freely available online. For that, this paper aims to enrich the Arabic content. The introduced dataset contains a large amount of data allowing users to use it with different models. Also, the dataset was cleaned to be ready to be used by other researchers. Finally, the dataset can be accessed freely.

4 Data Preparation

Table 6. Modern standard Arabic dataset categories distribution

Category	Text files
Art	300
Economy	300
Health	300
Law	300
Literature	300
Politics	300
Religion	300
Sports	300
Technology	300
Total	2700

4.1 Data Collection

The data was scraped and collected from two of the most famous news sources in Saudi Arabia. The selected sources were chosen since news are added more frequently. Also, due to the limitation of time only two data sources were selected, which are:

- Saudi Press Agency (SPA) [12]
- AlRiyadh Newspaper [13]

The tool used to scrape the data is ParseHub [14]. It is a cloud based free web scraping tool extracting up to 200 rows in each run. ParseHub tool is easy to export data from any website using the csv format. The collected data consists of news titles and news text details. Moreover, the news are categorized into six classes: Political, Economical, Sports, Arts, Social and General news. The dataset contains 45,936 records. More details about the categories and their totals are provided in Table 7 and Fig. 2. Furthermore, the data collection process lasted for several months, from May until October, 2019. The dataset is publicly available at https://drive.google.com/open?id=1uwD56jaVIbsQQWVqqyL08TgjuTraYFJC.

Table 7. Detailed classes of Saudi newspapers articles dataset

Resource	Political	Economical	Sports	Arts	General news	Social	Total
SPA	6,126	5,637	3,180	4,101	7,786	6,544	33,374
AlRiyadh	3,442	1,992	3,954	591	2,362	220	12,561
Total	9,568	7,629	7,134	4,692	10,148	6,764	45,935

4.2 Data Cleaning

The data was collected from different websites, therefore it required some cleaning to remove some unnecessary texts. For the data gathered from the Saudi Press Agency (SPA), the news details contained a header with the date of the news and the location. Also, at the end of the news article, it had a timestamp and the news URL, as shown in Fig. 3.The text in red shown in Fig. 3 (at right) was removed to be ready to use as displayed in the same figure at left. Besides, 1300 records had an empty news detail which was filled with the news title. Moreover, the news categories in the dataset was categorical (political, sports, etc.) which was converted to numerical values (0, 1, etc.) to facilitate its use in the classification process.

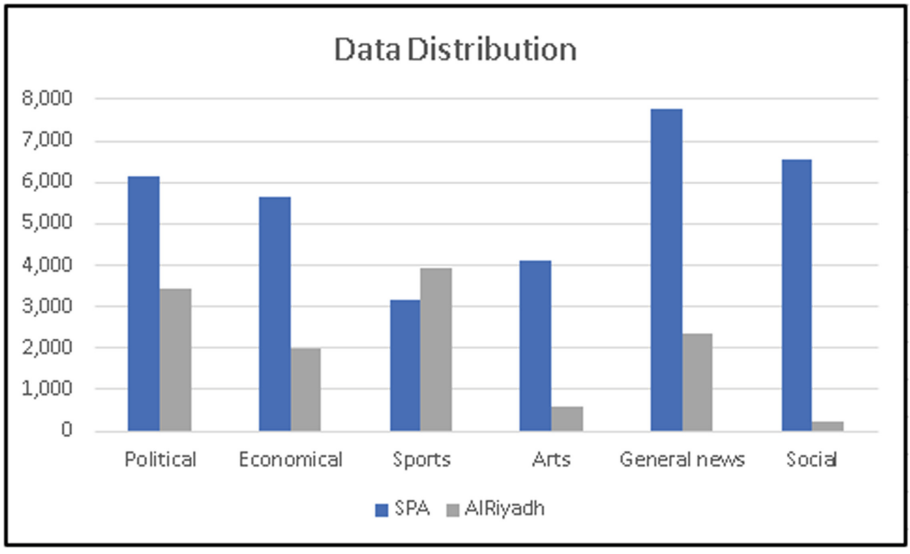


Fig. 2. Dataset distribution for different categories


القاهرة 16 شوال 1440 هـ الموافق 19 يونيو 2019 م واس
اهتمت كافة الصحف المصرية الصادرة اليوم بمباحثات القمة التي جرت أمس بين الرئيسين المصري عبد الفتاح السيسي والبيلاروسي ألكسندر لوكاتشينكو، وعكست توافقت وجهات نظر البلدين بشأن تعزيز التعاون على مختلف الأصعدة وفي مجالات التصنيع العسكري والصناعات الثقيلة.
وأشارت إلى أن السيسي ولوكاتشينكو اتفقا على ضرورة مكافحة الإرهاب والحل السياسي لأزمات منطقة الشرق الأوسط وتفعيل مجلس الأعمال المصري البيلاروسي ورفع مستوى اللجنة المشتركة لتصبح على المستوى السياسي.
وسلطت الضوء على تمكن قوات الدفاع الجوي لـ «تحالف دعم الشرعية في اليمن» من تدمير طائرتين مسيرتين تحملان متفجرات أطلقهما الحوثيون باتجاه المملكة.
وعرجت على إدانة الخارجية الفلسطينية لتصريحات وزير الأمن الإسرائيلي جلعاد أردان والتي حرض فيها على الإطاحة بالرئيس "أبو مازن" لرفضة مؤتمر البحرين الاقتصادي، مؤكدة أن التحريض يزيدنا إصراراً على إسقاط "صفقة القرن".
وأخيرت أن قوات الجيش الليبي تمكنت من القبض على عدد من المرتزقة الأجانب من جنسيات إفريقية تم جلبهم للقتال ضمن قوات حكومة الوفاق في معركة تحرير العاصمة طرابلس من المليشيات المسلحة والجماعات الإرهابية.
وذكرت أن الرئيس الأمريكي دونالد ترامب يبحث استخدام القوة العسكرية ضد إيران لحماية المصالح الأمريكية ومنع إيران من امتلاك سلاح نووي.
ونوهت صحف القاهرة بأن الرئيس ترامب استهل حملته الانتخابية أمس بتصريحات مثيرة للجدل حيث حذر من أنه سيتم طرد الملايين من المهاجرين غير الشرعيين خلال أيام، وأكد أنه يعرف من يقف وراء هجمات القاعدة الإرهابية في 11 سبتمبر 2001.
// انتهى //
09:47 م
0016
www.spa.gov.sa/1935868

اهتمت كافة الصحف المصرية الصادرة اليوم بمباحثات القمة التي جرت أمس بين الرئيسين المصري عبد الفتاح السيسي والبيلاروسي ألكسندر لوكاتشينكو، وعكست توافقت وجهات نظر البلدين بشأن تعزيز التعاون على مختلف الأصعدة وفي مجالات التصنيع العسكري والصناعات الثقيلة.
وأشارت إلى أن السيسي ولوكاتشينكو اتفقا على ضرورة مكافحة الإرهاب والحل السياسي لأزمات منطقة الشرق الأوسط وتفعيل مجلس الأعمال المصري البيلاروسي ورفع مستوى اللجنة المشتركة لتصبح على المستوى السياسي.
وسلطت الضوء على تمكن قوات الدفاع الجوي لـ «تحالف دعم الشرعية في اليمن» من تدمير طائرتين مسيرتين تحملان متفجرات أطلقهما الحوثيون باتجاه المملكة.
وعرجت على إدانة الخارجية الفلسطينية لتصريحات وزير الأمن الإسرائيلي جلعاد أردان والتي حرض فيها على الإطاحة بالرئيس "أبو مازن" لرفضة مؤتمر البحرين الاقتصادي، مؤكدة أن التحريض يزيدنا إصراراً على إسقاط "صفقة القرن".
وأخيرت أن قوات الجيش الليبي تمكنت من القبض على عدد من المرتزقة الأجانب من جنسيات إفريقية تم جلبهم للقتال ضمن قوات حكومة الوفاق في معركة تحرير العاصمة طرابلس من المليشيات المسلحة والجماعات الإرهابية.
وذكرت أن الرئيس الأمريكي دونالد ترامب يبحث استخدام القوة العسكرية ضد إيران لحماية المصالح الأمريكية ومنع إيران من امتلاك سلاح نووي.
ونوهت صحف القاهرة بأن الرئيس ترامب استهل حملته الانتخابية أمس بتصريحات مثيرة للجدل حيث حذر من أنه سيتم طرد الملايين من المهاجرين غير الشرعيين خلال أيام، وأكد أنه يعرف من يقف وراء هجمات القاعدة الإرهابية في 11 سبتمبر 2001.

Fig. 3. An article before (right) and after cleaning (left)

5 Data Validation

To validate the quality of the collected dataset (SNAD), the Convolutional Neural Networks (CNN) deep learning technique is applied. CNN is selected as it is known to perform well in processing the data with high dimensionality structure [15].

5.1 Data Preprocessing

After preparing the data, the preprocessing steps are employed as follows. First, cleaning the data from punctuation marks displayed in Fig. 4. Second, normalizing the dataset by removing digits, none Arabic characters, and removing diacritics. Third, removing stop words and tokenizing the texts. Finally, applying stemming to get the roots of each obtained token.

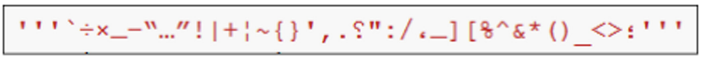

''' ` ÷ × _ – " … ” ! | + ¦ ~ { } ' , . ؟ " : / ، _] [% ^ & * ( ) _ < > ؛ ' ''

Fig. 4. Punctuation marks

5.2 Classification Results

The corpus is classified using CNN with a division of 70% for training, 15% for validation, 15% for testing. The training and validation sets are used to set the parameters of CNN. After several experiments (omitted in this article to save space), the parameters of CNN are set as follows.

– Ephocs = 40
– Batch size = 1000
– Optimizer: RMSprop

The training and validation accuracy rates reached 94.31% and 88.08% respectively. The Running time was 2:26:52 h. After finding the training model with the optimal parameters, the testing set is used. To confirm the efficiency of using the proposed dataset in classification, several metrics are used to evaluate the results as shown in Table 8.The classification accuracy reached a high value of 84% with a precision value achieving 85%. This explains that a large number of instances are correctly classified with a high precision. Moreover, the recall value indicates that almost 90% of the instances (text articles) are correctly classified into the associated class. Finally, the F1 score shows the balance between the recall and the precision reaching 86%. Consequently, the obtained results show that the proposed dataset is effective to be used in the classification purpose, especially in Deep Learning.

Table 8. CNN classification results for SNAD

Metric	Accuracy	F1 score	Recall	Precision
Results	0.8432	0.8584	0.8704	0.8499

6 Conclusion

This paper aimed to propose a new Arabic Text Classification dataset called Saudi Newspaper Articles dataset (SNAD). The dataset has a large size to be used in deep learning for text classification. It is collected from the Saudi Press Agency (SPA) and AlRiyadh Newspaper websites. It has six different categories with 45,935 articles. The dataset is cleaned and deeply preprocessed. It is validated using the CNN classifier reaching an accuracy of 84.32% which proved its efficiency in the classification purpose.

References

1. Zitouni, I.: Natural Language Processing of Semitic Languages. Springer, Heidelberg (2014)
2. Comrie, B.: The World's Major Languages. Routledge, Abingdon (2009)
3. Shah, M.: The Arabic language (2008)
4. Einea, O., Elnagar, A., Al Debsi, R.: SANAD: single-label Arabic news articles dataset for automatic text categorization. Data Brief **25**, 104076 (2019)
5. Boukil, S., Biniz, M., El Adnani, F., Cherrat, L., El Moutaouakkil, A.E.: Arabic text classification using deep learning technics. Int. J. Grid Distrib. Comput. **11**(9), 103–114 (2018)
6. Alalyani, N., Marie-Sainte, S.L.: NADA: new arabic dataset for text classification. Int. J. Adv. Comput. Sci. Appl. **9**(9) (2018)

7. Belkebir, R., Guessoum, A.: TALAA-ASC: a sentence compression corpus for Arabic. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1–8. IEEE (2015)
8. Abuaiadah, D.,El Sana, J., Abusalah, W.: On the impact of dataset characteristics on Arabic document classification. Int. J. Comput. Appl. **101**(7) (2014)
9. Sabbah, T., Ayyash, M., Ashraf, M.: Hybrid support vector machine based feature selection method for text classification. Int. Arab J. Inf. Technol. **15**(3A), 599–609 (2018)
10. Abuaiadah, D.: Arabic document classification using multiword features. Int. J. Comput. Commun. Eng. **2**(6), 659 (2013)
11. Alhawarat, M., Hegazi, M.: Revisiting K-means and topic modeling, a comparison study to cluster arabic documents. IEEE Access **6**, 42740–42749 (2018)
12. The official Saudi press agency, May 2019. https://www.spa.gov.sa/
13. Alriyadh newspaper, May 2019. http://www.alriyadh.com/
14. Parsehub, May 2019. https://www.parsehub.com/, May 2019
15. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)

Evaluating Deep Learning Biases Based on Grey-Box Testing Results

J. Jenny Li(✉), Thayssa Silva, Mira Franke, Moushume Hai, and Patricia Morreale

Kean University, Union, NJ 07083, USA
juli@kean.edu

Abstract. The very exciting and promising approaches of deep learning are immensely successful in processing large real world data sets, such as image recognition, speech recognition, and language translation. However, much research discovered that it has biases that arise in the design, production, deployment, and use of AI/ML technologies. In this paper, we first explain mathematically the causes of biases and then propose a way to evaluate biases based on testing results of neurons and auto-encoders in deep learning. Our interpretation views each neuron or autoencoder as an approximation of similarity measurement, of which grey-box testing results can be used to measure biases and finding ways to reduce them. We argue that monitoring deep learning network structures and parameters is an effective way to catch the sources of biases in deep learning.

Keywords: Neural Networks · Deep learning · Mathematical interpretation · Deep learning evaluation · Bias measurement

1 Introduction

Neural Networks (NN) with multiple layers of perceptron and deep learning (DL) with multiple convolutional layers are currently the two mainstream techniques for AI machine learning because they are successful in performing many machine-learning tasks, such as image recognition, speech recognition, and language translation. Of particular interest with this paper are AI machine learning systems that sort and classify people according to various attributes, as are used in algorithms involved in criminal sentencing or hiring and admission practices [1]. Studies found that the algorithms now in use to automate the pre-trial bail process unfit to do so, as they suffer from data sampling bias [2] and bias in statistical predictions, as well as human-computer interface issues and unclear definitions of high risk and low risk [3].

It remains a pressing open problem to understand mathematically where biases come from and how to measure them [4]. Understanding biases [5] of deep learning has many benefits, including attribution of accountability, better design of NN, determination of the unbiased design of DL, and improvement in their applications. Many NN or DL systems are implemented using existing tools, such as TensorFlow [6], Keras [7] and PyTorch [8], where the white-box testing of the source code is not feasible. On the other hand, the full black-box testing of such systems has the issue of not being able to cover

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 641–651, 2021.
https://doi.org/10.1007/978-3-030-55180-3_48

all possible scenarios to be encountered by the system. Therefore, the grey-box testing to validate a NN or DL structure is the most suitable for ML-based systems. It at least compliments other testing techniques for the evaluation of ML-based systems. In the grey-box testing of an ML-based system, we look into the **neuron structures and their parameters** of a NN or DL to calculate their potential biases.

One of the most prominent success of NN was Hinton's work on the recognition of one thousand images using six million training images in 2012 [9]. Since then, many researchers have attempted to interpret NN or explain how and why it works. Subsequently, DL became more popular as reported in [10, 11] with eye-catching success stories. We tried to explore the mathematical explanations of the success of NN and DL, using our interpretation of each neuron in NN and the convolutional layer in deep learning. We then use the interpretation to measure biases based on grey-box testing results.

2 Interpretation of Deep Learning

Currently there are two schools of NN and DL interpretations. Traditionally, each neuron/perceptron is considered to be a node taking the accumulated effect of each decision-impacting factor with synaptic weights as input. The input factor with a higher weight value has more impact on the output decision of the node. The second more recent interpretation of DL uses topological manifold concepts to explain encoders of a convolutional layer as transformers to lower dimensions while keeping manifold structure homeomorphism. We will explain the two existing interpretations and discuss how to use them for the measurement of biases.

2.1 Interpretation 1: Neural Multipliers are Weights

A neuron is traditionally defined as a function of $z = f(X, W, T)$, where X is a vector of the input factors, W is a vector of multipliers with one value for each input factor, and T is the threshed for output decision of different categories, which was later replaced by a sigmoid generation function [12]. Assuming that the desired output is $d = g(X)$, then the training of a neural network is to maximize the performance function of negative error, $P = -|d - z|^2$. This definition suggests some kind of regression to calculate the weights for each contributing factor so that the output will be the closest to the desired one.

To validate the conventional interpretation of neuron multipliers as contributing weights of output decisions, we designed grey-box test cases to modify the multipliers to be the values within a scale. In a grey-box testing, we do not have the access to the source code, but we can examine and modify neuron structures, i.e. we adjust the multipliers to percentage values. Since the overall contribution of all weights should add up to one, the summation of the multipliers should also be one with each multiplier as the percentage of the weighted contribution. If the multipliers are indeed weights, they should be able to scale proportionally and the usage of the percentage values of all weights should not affect the performance of the neuron.

In our testing, the neural network with scaled weights consistently performed worse with higher error rates. Every time when we adjusted the multiplier to be a percentage weight that adds up to one, the neuron failed to perform. This observation suggests that the multipliers of each neuron **are not weights that are scalable**. They are approximation values.

2.2 Interpretation 2: Learning of Manifold Structure

The second type of interpretation considers deep learning as the learning of manifold structure where the space transformation [13] through autoencoders (and decoder) keeps the homeomorphism of the manifold structures [14].

To validate this type of interpretation, we use deep learning implemented in TensorFlow to be trained with chemical component structures and regenerate new ones. Again, we used grey-box testing where we can monitor NN or DL structures and parameters, but not their source code. If the new space after the autoencoder transformation is indeed the reduction of the original space, then the regenerated structure should be homeomorphic with the original one. However, our testing results demonstrated that the generated structures were not homeomorphic with the original ones, which seems to suggest that the interpretation of manifold structure learning might not be complete. Depending on the decoder, the output might not be in a homeomorphic space. In fact, the autoencoders of a convolutional layer abstracts the meaning of the structure of the training data and the decoder could potentially move to a different domain using a different abstract.

2.3 Interpretation 3: Neuron/Autoencoder as *Similarity* and Bias as the Difference

In our interpretation, both neuron and autoencoder approximate the **degree of similarity** between the input factors and the stored or entered factors of the target feature through **a rough calculation of their square differences**. The multipliers of the neuron inputs are not weights. They are actually a target feature of fixed values, which does not scale proportionally and does not add up to one. The autoencoder of deep learning is also an approximation of the similarity between the input and the target images roughly calculated through their differences. The higher the value is, the less difference between the two images and the more similar they are. The biases are the difference between the multipliers and the actual inputs or between the feature and the actual inputs. If multipliers or features do not have biases, then the NN or DL system reflects the correct situations.

Each Neuron and Autoencoder Approximates the Similarity and the Biases are Their Difference from the Desired Models. This interpretation is also consistent with how human thinks. Human uses image or situation similarity to make decisions. Identifying degree of similarity is the foundation of many of human cognitive abilities. It also explains biases. The multiplier values or features depend on the training set and its difference with the actual input is the bias.

The convolutional layer of deep learning includes encoder for filtering, pooling to select maximum, and normalization to remove negative numbers, as well as fully connected layer to calculate the probability for each output category. Like neurons,

the autoencoders compare the similarity of the input vector with a target feature. The pooling selection of the largest similarity value is in fact the selection of the best fit to the feature, which seems reasonable. However, the normalization of removing the negative similarity might not be necessary because negative means completely different from the target feature and such information could be useful to detect untrained patterns in some applications such as cybersecurity attack detection. With this interpretation of neurons and autoencoders as similarity calculator, we can now explain the characteristics of neural networks and deep learning, as well as measuring biases based on testing results through monitoring of deep learning network structures and parameters. We can also calculate the optimal designs of neural networks and the optimal number of convolutional layers, leading to quicker training algorithms.

3 Theoretical Explanation of Bias Sources

First with the similarity interpretation and the bias definition and measurement, we can explain the following scenarios of NN and DL and the causes of biases.

Question 1: Why does more training produce better results?

As presented by Michael Belkin in [15], they discovered that when the training of NN continuous and keeps going, the fit gets better, i.e. so called "double deep phenomena". Since the neuron multiples are the storage of the vectors, to which the input vector is measured against for similarity, the multiplier vector is representing all the training inputs to that neuron. The more training means more data being counted into that representation. When more data are included, it will eventually converge to the "correct" answer, which may consist of biases. To minimize the absolute error between the representation and the data set, we can use their median value as the representation and to minimize the square error, we can use their mean. More training will help the system to eventually converge to the median or the mean of the training set. The biases come from the drifting of the median/mean when the input samples are not truly representative.

The highly cited paper by Ben Recht [16] also stated that randomly generated data could be fit essentially exactly with enough training. The explanation is the same as the above paragraph. The random data are most likely to represent a normal distribution. The more training, the more likely for it to converge to the mean or medium of the training data set.

Question 2: Why does not increasing the number of layers of a multi-layer perception neural network improve its performance? The performance seems to plateau after five or six layers.

In multiple-layer neural network, the input to the next layer is the output of the previous layer. Since each neuron approximately calculates the similarity/difference between the input and the W multiplier, the input to the next layer is actually a calculation of the difference from the previous layer. For example, the second layer input is the calculation of the difference of the first layer and its output is the difference of the difference. Similarly, the third layer calculates the difference of the difference. When we move on to the fourth layer and so on, it calculates the difference of difference, etc.

As it can be seen from this pattern of continuation, the difference of differences will eventually be exhausted out. Therefore, we can see that the performance of a neural network is maximized at certain number of layers and does not improve or even gets worse after certain number of layers of perceptron.

This raises another question of where the maximum boundary is and how to design an optimal neural network. We addressed this question in another report due to the space constraint of this one.

Question 3: Why does reducing the number of neurons on each layer sometimes actually improve its performance?

Experimental research shows that "dropout techniques" improves performance. Here is the explanation. Each neuron roughly calculates the difference between the input and the desired test subject. The number of subjects on that layer determines the need of each layer. More neurons than are necessary given the number of test subjects will not affect the output. Often replicate neurons may arise due to imprecise training of the NN.

For example, as discovered and reported in [17], dropout tends to make the norm of incoming/outgoing weight vectors of all the hidden nodes equal. The explanation is that the dropout includes the neurons that do not represent any subject and are redundant one that should be removed. The weight vector becoming equal means that there is only one subject to be compared with and thus only one neuron is needed at that layer. This confirms our similarity interpretation of neurons.

On the other hand, if the neuron number were too small, not all subjects to be compared with would be included in the network for recognition, and then the performance of the NN would suffer as well. There is a certain number of neurons on each layer for a NN to achieve the best performance. Missing neurons may also cause biases in the decision made by the network.

Question 4: Why do neural networks have impressions? i.e. once NN learn a pattern, it is difficult to unlearn.

Contrary to popular understanding, each neuron is a storage unit of a target pattern that the input will be compared to, checking for similarity. Because the neuron is acting as static storage, an entire NN will tend towards impressions because each neuron will not change its stored pattern and a NN is just a construction of layered neurons, each of which is difficult to change. This also explains biases. If the impression is not the desired ones for the users, then it has biases, which comes from the multiplier values, which can be obtained through grey-box testing.

Question 5: Why does deep learning with a controlled number of autoencoders perform better than multi-layer neural networks?

Deep learning with a controlled number of autoencoders only includes a precise number of layers. However, multi-layer NNs often have redundant layers of neurons because they are randomly assembled by trial and error. These extra layers may even create deviations from the accurate output.

Additionally, autoencoders compare inputs to dynamically entered features while each neuron in a multi-layered NN uses a predefined pattern, which is stored in its multipliers to compare with the input, making multi-layered NN less versatile in recognizing

patterns. Features represented by autoencoders are changeable to remove biases, while it is difficult to change NN once trained.

Question 6: Why does Radius Neighbor (the method we invented in 2016 [18]) perform better than multi-layer neural networks and with drastically reduced training time? Will it also be better than deep learning?

Radius Neighbor uses a mathematical approach to identify the multipliers of a neuron without the need for gradient descent, which reduces the training time significantly. In addition, it uses the optimal design of NN structure based on the similarity interpretation of the meaning of each neuron; thus, its performance should be comparable to that of deep learning. Their comparison study through testing is ongoing.

Question 7: Why does a convolutional neural network [19] only capture local "spatial" patterns in data? If the data does not look like an image, it is less useful.

The autoencoder of the convolutional layer uses a feature vector to compare to a window of an image. Because for images, the data closer together are more related. For example, image data within a window are more likely to form a feature than the data from one end of an image to the other end. The traditional way of scanning images row by row does not work well because data of each row are less related to data of each window, which are close together. Therefore convolutional neural network should work for any data of which window features can be identified, regardless of the source of data whether are from an image or not.

Question 8: Why does deep learning have such a rule of thumb: If your data is just as useful after swapping any of your columns with each other, then you cannot use convolutional neural networks.

The reason why the convolutional layer does not work for swappable data is consistent with the similarity and bias interpretation for two reasons. First, the position of a factor in the vector affects the similarity calculation. Supposed the feature vector is <W1, W2> and the input vector is <X1, X2>. Swapping X1 and X2 will totally mess up the similarity calculation. Secondly, when the rows are columns are switchable, the data close in the position of the table are not necessary more related to each other, and thus might not work using the window features. Therefore, switchable columns or rows make it difficult to find an appropriate feature for similarity and bias measurements. It is unimportant if columns are switchable. The key of the convolutional layer is to arrange the data in a way that features are certainly identifiable.

Question 9: Why are convolutional neural networks so great for finding patterns and classifying images?

As explained above, the convolutional layer compares windows of an image to features. As long as data in closer positions are more related and features are identifiable, it should always work well.

Through answering the above nine questions of deep learning networks, we identify the network structures and parameters as one source of biases. The next section we explain how to design networks with less biases and how to test for biases using a grey-box approach of monitoring network structures and parameters.

4 Design and Testing of (Convolutional) Neuron Network Structures and Parameters

The similarity and bias measurement of neurons and autoencoders can help to design optimal networks, decide the number of convolutional filtering and pooling layers in deep learning, and validate them through grey-box testing of monitoring W multipliers and autoencoder feature vectors.

We start with a simple example of designing a neuron to recognize AND gate operation. We want the neuron to have the maximum similarity with the input pairs of (0, 1), (1, 0), and (1, 1) to generate the output of "1". And the input pattern similar to the pair of (0, 0) should generate the output "0". For the convenient of discussion, we assume that our neurons use a threshed to decide the output rather than the sigmoid function. The discussion here should also work for sigmoid situation.

We now convert the AND gate problem to similarity problem that can use neurons. To decide if an input is similar to any of the three, we need to minimize the error between the input and the three training patterns. Mathematical prove of the minimization is straightforward. To minimize the absolute error, we can use their median and to minimize the square error, we can use their mean. Therefore, to check if an input is similar to the three pairs, we can use the mean of 1/3 and 1/3 on each input. We now create a neuron that can recognize patterns of producing an output of one. Figure 1 shows this neuron. The threshed for the final output can be anything between (0, 1/3), from which we pick 1/6 to maximize the margin to be 1/3. The goal of maximizing the margin is to avoid overfitting.

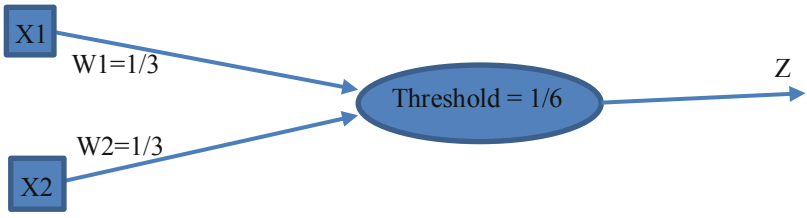


Fig. 1. An optimal design of a neuron to recognize an AND gate.

Now the question is how to calculate the biases based on the testing results of the NN in Fig. 1? As stated previously, the grey-box testing monitors the values enter into each neuron/autoencoder and the biases are calculated as the difference between these values and the multipliers or the features. We were able to prove the correctness of this formula mathematically.

We use another example to demonstrate the design and testing of NN and DL systems. Classification and recognition of an XOR function is a classical problem [20] for machine learning to demonstrate various learning algorithms such as kernel tricks [21], Support Vector Machines, Radial Basis Functions and neural networks. Our Radius Neighbor method derived based on similarity interpretation of neurons can also quickly construct a neural network with multipliers calculated mathematically without the need of going through gradient decent training.

In XOR, the input pairs of (0, 1) and (1, 0) produce "1" as the output and the pairs of (0, 0) and (1, 1) generate output "0". Again, we use the two cases of generating output "1" as similarity patterns to create a neuron to roughly calculate the similarity between the neuron input and the two pairs of (0, 1) and (1, 0), as shown in Fig. 2.

The above two design examples shows that the biases come from the training set and are embedded in the network structures and parameters, which can be detected through grey-box testing.

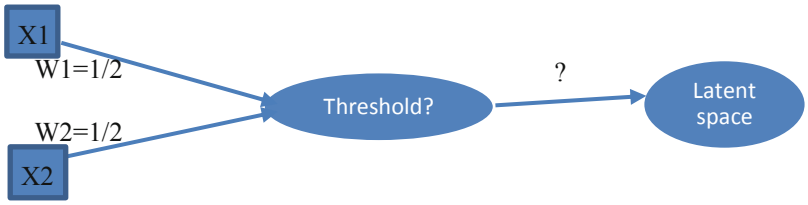


Fig. 2. An initial neuron for partial XOR.

At this point, only the multipliers of one neuron are calculated. The next step in the NN construction is to feed all training cases to this one neuron to generate their output values. All output values are now data of one dimension to be separated into various object categories recognized by the neuron. For the convenience of visualization, we put all output values from the neuron on an X axiom line and mark them with output categories, as shown in Fig. 3.

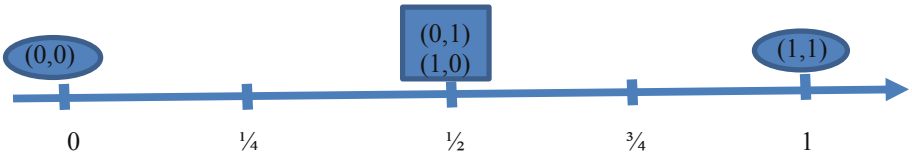


Fig. 3. A linear segmentation line with more than two segments.

The current neuron threshed approach or sigmoid function can only separate the one dimension output linearly into two categories. Figure 3 shows a situation when the outputs on a line have more than 2 segments. Therefore, no threshed or linear divider can be determined and thus another layer of neuron is necessary in this case.

The second layer of neurons now have three inputs, the pair of the original input plus the additional similarity information from the first layer. For the pairs of (0, 1) and (1, 0), we still want the similarity to be as close to them as possible and thus the multipliers of ½ and ½ will still be used for them. In addition, we now also have a similarity measurement as the output of the first neuron to be fed into the second layer as an additional input.

To separate the second segment away from the third one with the maximum margin, we select the middle value between ½ and 1, which gives us the value of ¾ to be used as the threshed of the first neuron. With this threshed, we now shift the output of the

similarity from the first neuron by "−3/4", i.e. each of the two pairs have the difference of -3/4 from the first neuron. Again, we want the second neuron to be the most similar to both (0,1) and (1,0), as well as the difference of −3/4. Therefore, the second neuron should have the multipliers of original two ½ with an additional of −3/4 for the output from the first neuron. In two steps with the computation of O(1), we obtain a neural network for interpreting XOR as shown in Fig. 4.

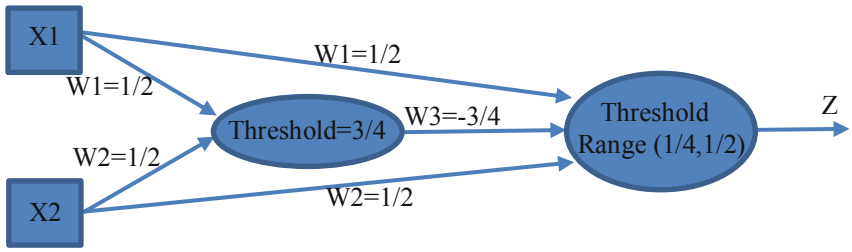


Fig. 4. A linear segmentation line with more than two segments.

Based on the above example, the computation intensive step of network training, gradient decent, is not used anymore and an optimal network design based on the training data can be created algorithmically. It also seems that there is a lower bound on the number of neural layers needed for each training set. We define a concept of Linear-Segmentation (LS) to help with the determination of the number of neuron layers and number of neurons on each of them.

We found that the value of LS determines the lower bound of the number of convolutional layers.

Definition: Linear-Segmentation (LS) counts the number of segments on the linear line of the output of a neuron before its sigmoid function.

Lemma 1: The lower-bound of the number of layers needed for NN or DL is the value of LS minus 1 for neurons and LS minus 2 for autoencoders of DL.

The above examples describe our Radius Neighbor algorithm for creating and training NN and DL convolutional layers without the usage of gradient descent, based on the similarity interpretation of neurons and autoencoders.

We applied the above-mentioned deep learning networks to object recognition of videos streamed from drones. More specifically, we use drone video streaming to automatically identify open parking spaces. We then use the grey-box testing to validate such a ML-based systems. Besides checking the correct identifications, we also calculate the biases based on values collecting from the difference between the input and the desired open space feature. Due to the space constraint, we will report these testing results in another paper.

5 Concluding Observations

We discovered an interpretation of neurons and deep learning autoencoders and used them to explain NN characteristics and DL convolutional layers, as well as calculating

biases through grey-box testing of monitoring neuron structures and their parameters. This interpretation helps us to better understand NN and DL to generate new research topics with the potential of improving neural network-based machine learning and their testing. We used an orthogonal array to validate the coverage of the testing set [22].

We use the interpretation of neurons, autoencoders and biases to explain all characteristics of NN and DL that we have encountered so far. We also use it to create an algorithmic approach for training neural networks with the potential of drastically reducing time and resource requirements. The bias calculation through grey-box testing provides another layer of assurance for ML-based systems. Overall, this work is promising in achieving groundbreaking results by providing methodologies to design optimal NN and DL convolutional layer and to train and test the network to complement the computationally intensive gradient decent based training methods.

Beyond investigating mathematical interpretations and testing, many more research topics are enabled by this research, such as the invention of new kind of neurons and their effectiveness testing with less biases [23], the exploration of learning evolution and relearning, the definition of effective networks, the identification of upper bound of neural layers for optimal performance, and the investigation of hierarchical multiple concurrent neural networks for human brain simulation. The results of this work of testing ML-based systems also helped us realize that NN and DL are a closer resemble of human brain than we had expected. Through this work, we hope that we contribute to the understanding and testing of human brain simulation using neuron networks.

Acknowledgments. We would like to thank Kean STEMPact program for supporting us to conduct this research in the summer of 2019 and Google's subsequent support through a TensorFlow research grant.

References

1. Whittaker, M., Crawford, K., Dobbe, R., Fried, G., Kaziunas, E., Mathur, V., West, S.M., Richardson, R., Schultz, J. and Schwartz, O.: AI now report 2018. AI Now Institute at New York University (2018)
2. Binns, R.: Fairness in machine learning: lessons from political philosophy. In: Conference on Fairness, Accountability and Transparency, pp. 149–159, January2018
3. https://www.partnershiponai.org/. Accessed January 2020
4. Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E.P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 329–338. ACM, January 2019
5. Corbett-Davies, S., Goel, S.: The measure and mismeasure of fairness: a critical review of fair machine learning. arXiv preprint [CS] arXiv:1808.00023, 31 July 2018
6. https://www.tensorflow.org/. Accessed January 2020
7. https://keras.io/. Accessed January 2020
8. https://pytorch.org/. Accessed January 2020
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, pp. 1097–1105, 03–06 December 2012, Lake Tahoe, Nevada (2012)

10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
11. Feng, X., Zhang, Y., Glass, J.: Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1759–1763, May 2014
12. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Press, Upper Saddle River (2009)
13. https://nlp.stanford.edu/pubs/clark2019what.pdf. Accessed January 2020
14. Lei, N., Su, K., Cui, L., Yau, S.-T., Gu, D.X.: A geometric view of optimal transportation and generative model. arXiv: 1710.05488. https://arxiv.org/abs/1710.05488. Accessed November 2019
15. https://www.youtube.com/watch?v=5-Kqb80h9rk. Accessed November 2019
16. https://arxiv.org/abs/1611.03530. Accessed November 2019
17. https://arxiv.org/pdf/1806.09777.pdf. Accessed November 2019
18. Li, J.J., Rossikova, Y., Morreal, P.: Natural language translator correctness prediction. J. Comput. Sci. Appl. Inf. Technol. **1**(1), 2–11 (2016). ISSN Number 2474–9257
19. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
20. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
21. Bengio, Y., LeCun, Y.: Scaling learning algorithms towards AI. In: Bottou, L., Chapelle, O., DeCoste, D., Weston, J. (eds.) Large Scale Kernel Machines. MIT Press, Cambridge (2007)
22. Hedayat, A.S., Sloane, N.J.A., Stufken, J.: Orthogonal Arrays: Theory and Applications. Springer, Heidelberg (2012)
23. Pryzant, R., Richard, D.M., Dass, N., Kurohashi, S., Jurafsky, D., Yang, D.: Automatically neutralizing subjective bias in text. https://arxiv.org/abs/1911.09709. Accessed March 2020

Novel Deep Learning Model for Uncertainty Prediction in Mobile Computing

Anand S. Rajawat[1], Priyanka Upadhyay[2], and Akhilesh Upadhyay[3](✉)

[1] Department of CS Engineering, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore 453111, Madhya Pradesh, India
rajawat_iet@yahoo.in

[2] Department of Applied Science, SIRT, SAGE University, Indore 452020, Madhya Pradesh, India
ramrajaacademy@gmail.com

[3] Department of EC Engineering, SIRT, SAGE University, Indore 452020, Madhya Pradesh, India
akhileshupadhyay@yahoo.com

Abstract. Mobile phones have become one of the most common and essential tools of humans life. What initially started as a basic hand model with telephone like features has developed into a mini computer that has all our personal and professional life stored into it. But is this information protected and safe? Hacking and using our information has been a common problem for a long time. In case of websites, we have several tools and applications to protect our computer from hacking and blocking the advertisements that can track our computer. But there are no tools like this available for mobile applications. We install several applications and provide access to all our information stored in our phone. It is the duty of the operating system makers like Google (Play Store) and Apple (App store) to create proper guidelines for the applications to protect our privacy, and enforce them. But with the increasing number of applications and its usage, not everything is policed properly. For example, even though applications installed in apple iPhone requires the permission of users to access and transfer the data, the apple does not intervene and check these applications if they are accessing only the permitted data or not. This research paper concentrates on developing an algorithm to prevent and detect uncertainty in these applications. The algorithm constructed is based on deep learning model, which is a consolidated tool used to predict the uncertainty in mobile computing. The challenge of calculating this uncertainty estimation is done by learning both target output and its corresponding variance. Then a similar estimation is performed in the newly constructed result model which contains one network for target output and another for variance. This method prevents error occurrence at a higher percentage than other algorithm available.

Keywords: Deep learning · Uncertainty · Convolution neural network · Mobile computing

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 652–661, 2021.
https://doi.org/10.1007/978-3-030-55180-3_49

1 Introduction

When mobile phones were initially developed they were simply basic modeled phones which were technically landline telephones that can be taken everywhere. The biggest revolution in the world of mobile phones came when Google introduced Android Operating system in 2008. From then the growth of Android phones have been enormous, taking over the whole world and also exceeding iOS and Windows system along the way. The main reason for this growth and success is because of the number of applications that were developed based on the android operating system. Applications like Facebook, Gmail, Chrome, YouTube etc. which were available only available in computers were remodeled to run in Android operating system. Based on this many other applications were also developed. Now users can download applications for everything from ordering food to booking tickets, watching movies, etc. The applications are available for download from either original Google market or third party market. These applications besides attracting several users also have gained the attention of Malware developers who make some applications that can obtain private information from the mobile devices. And it would depend upon the settings of his application. The truth is the data and everything in our phone can be tracked and hacked based on the applications and its settings in our phone. With the development of technology and its applications, several applications are being downloaded every day. People download without knowing the functioning of the applications. Each and every application is required to get the permissions of the users before accessing their information. But the fact is that even though the permission is obtained, there is no saying how much data is extracted, and how it is used. There is no one checking or guarding these data. A recent report showed that several Trojan infected applications were not detected during the normal Google checkup and some popular applications were already downloaded by several numbers of people before it was detected and removed, in the year 2018 Google analyzed and announced the types of malicious applications available. These details were updated in Google Play. Even Google's very own Google protect software to destroy these malwares may sometimes result in failure due to variations. The malwares are divided into four types namely,

- First is malwares that hack the phone, steals the data available and destroys the device in the process.
- Next is Spyware which steals the important private data like phone numbers, bank id's passwords etc. and moves it to others.
- The last two are Grayware and Adware that inserts various unwanted advertisements or popup in between the programs while it is running.

Then this model is inserted into our application which can detect the malware in other applications effectively. The initial or the traditional method that was created to evaluate these malwares is now useless in front of the new technology malwares that have been created. With the updating of the malwares the detecting methods must also be upgraded and more efficient than them. The continuous research in this field gave rise to an automatic mechanism which is trending right now in the field. The deep learning or the machine learning models (multi-level Convolutional Neural Network) are being used increasingly to create the algorithm to detect these new generation malware. The paper

is classified as: Sect. 2 analyses the researches and studies that had previously been done on this idea uncertainty prediction; Sect. 3 describes research analysis of related work; Sect. 4 analyses the problems and provides a new technique for detecting the malwares by creating a model based on CNN or convolutional neural network and fusion based network; Sect. 5 describes the method in which the data for feeding into the model is prepared and perform the evaluation of deferent number of machine learning algorithm, and how the model is trained and evaluated and finally the conclusion is given in Sect. 6.

2 Uncertainty Prediction

A recent study found that there was data exchange available between the android applications in the same phone. Two applications in the same phone can sometimes share information, which can be leaked to another source. It called as inter component communication or ICC. Here one application which had been permitted to access the phones location can access it and share it with another application in the phone, which can send the data to an unknown external server. The application uses ICC to communicate among themselves [3, 4]. Continuous researches are being made in this field to develop an antidote for detecting these malware. Various methods have been suggested to detect these malwares. Some of the methods include Static analysis, Dynamic analysis and Hybrid analysis. Static analysis is a light weight method used to detect abnormal signatures, functions in the source code. It detects the malware without really running the application. Though this method of analysis is faster compared to others it cannot be run on powerful applications that can hide or freeze themselves during examination to pass over. To overcome the disadvantages of static analysis, dynamic analysis was found. This analysis runs in the real device Table 1: Features set for analysis and evaluates the application directly to detect the network traffic and the application is examined while running to detect the errors and various malwares available. This procedure is time consuming compared to static analysis. It also occasionally results in large overhead. To overcome the disadvantages of both these methods, the researchers combined them to produce hybrid analysis, which provides a higher performance comparatively.

The main aim of this study is to prevent our data from being extracted or hacked by building a Convolutional Neural Network that can predict the uncertainties in mobile computing. The uncertainties are initially estimated and calculated and then using it in a newly constructed model will help us detect these data extraction and prevent them from getting hacked. Deep learning model otherwise known as hierarchical learning is a most recent method used to build and train neural networks, by learning the representation data available. It is nothing but a collection of high ended algorithms that could train and produce neural networks. Neural networks are nothing but decision making nodes, and the networks built using deep learning model are of higher quality compared to other neural networks. Deep learning is a part of machine learning family, but unlike other machine learning algorithms whose accuracy stops at a particular level, accuracy of deep learning will never reach a end. The accuracy grows with the amount of data you feed them. The more data you feed them the higher the accuracy. Now days with everyone using internet, we have access to large amount of data. So using deep learning model, we can easily train neural networks to provide better performance.

Table 1. Features set for analysis

Static analysis		Dynamic analysis	Hybrid analysis
Required permission	Sensitive API	Dynamic behavior	Higher performance comparatively
Camera, contact, voice recording, CPU usage, media, GPS, SMS reading, profile account linking, cookies, profile hack, reading mail, read location	Background process Application usage, Timely behavior, read back ground process behavior Shutdown, Restart package	Load Application Application usage, Timely behavior, read back ground process behavior Kill Background process, Service restart	Camera, contact, voice recording, CPU usage, media, GPS, SMS reading, profile account linking, cookies, profile hack, reading mail, Load Application Application usage, Timely behavior, read back ground process behavior

Convolutional and recurrent neural networks which are the products of deep learning model are highly popular and used in several applications like, computer vision, speech recognition, bio informatics etc. The only problem with deep learning neural networks is they are poor at collecting predictable uncertainties, and believe that their predictions are perfect. So in order to solve this problem the uncertainty output prediction is done by learning the targeted output data and its corresponding variance. The uncertainties are available in different varieties for different applications. It is a vast area of data. But vaster the data is, more accurate the output will be. The data of several target outputs and is corresponding variables are fed and learned through the deep learning model and then a target output and variance is calculated. These are then fed into a final model and the uncertainty is calculated by reducing the error to a greater extent. The final model contains one network for target output and another one for the variance. This model prevents both Aleatoric and Epistemic uncertainty from occurring. Aleatoric uncertainty or statistical uncertainty provide different unknown every time we run it. Epistemic or systematic uncertainty is what is known in principle but not in practical application. Both these uncertainties can be overcome using this model with the help this model, with its expanding feature. This model is then used in our mobile applications to detect the uncertainties occurring in our applications. The several uncertainty data collected from other models and application are already fed into our model. So the when these uncertainties are occurring in our applications they can be easily detected and resolved. The models are trained to detect any new form theft scheme also. They are also found and the data of his new hacking methods will also be stored into our model and used for future applications. Thus this model algorithm using deep learning method will adapt and grow, continuously providing results and also reducing errors simultaneously. The chief objective of this paper is that detect these malware by using a deep learning model. The static features of various applications are collected and then they are organized in strings and converted into vector based data and fed into the deep learning model. The

model/algorithm will analyze the data and learn the combination and structure of the malware in the process.

3 Related Work

Zegzhda, P. et al. [1] provided a method to detect the malwares by using the deep learning model, most commonly called as CNN or convolutional neural networks. Here the design of the android application is used to find the malwares or other malicious applications. Instead of constraining ourselves to one particular type, here they have observe and study on all types of malicious applications available. Backes, M. et al. [2], used a Bayesian based pipeline method along with LUNA which is used to detect the uncertainties in the applications. The datasets are collected using the idea and tools of probabilistic programming. Then a model that protects the uncertainties is created to apply a large quantity of dataset that contains the details about the malware into the machine learning model. The results are then analyzed along with the parameters. Peiravian, N et al. [3] by merging the API calls with permission, the functioning of the applications are received through machine learning.

Droid-Sec [4] in order to detect the malicious applications, he suggested the usage of deep learning models since traditional models were less effective in finding the malicious program. He called it deep android malware detection [5]. With the failure of the traditional approach of the android detection a much better solution with the usage of CNN Zhang, Y et al. [6] which is comparatively faster than the traditional methods like linear –SVM and KNN was introduced. Yeh, C.-W. et al. [7] by merging both property along with flattened data set, the dimensions are decreased using K-skip-n-gram technique. This method allows learning both flexible and complex malwares data. Canfora, G., et al. [8] his method involves collecting the data of resources that has been used by the application since hiding these would be far more difficult.

4 Proposed Methodology

SVM and Linear SVM that follows old methods have poor structural composition with the usage of less computation unit layers. This was overcome with the development of deep layer technology. Training can be done using various approaches and methods. Deep learning model [9] because of its expanded structure any kind of method or algorithm can be used to construct and train it. Since the model consists of both learning and training, two levels are implemented. The first level is unsupervised pre-training phase in which the back propagation neural network are bundled together to construct the DBN or deep belief network. This is used for determining the character of android applications. The second level is Back propagation, the network from previous level is refined using labeled values under supervision. The deep learning model is constructed perfectly after this level. The model consists of a merging network called multi-level Convolutional Neural Network or MLCNN which is made up of three components:

- Accentuate classification component: the applications are subjected to a wide range static analysis and the data is taken from Android Manifest xml files and disassembled Dex files.

- Accentuate interleaving component: the data sets of various dimensions are joined together in a single vector space in order to increase the feature space.
- Discover component: multi-level CNN is used for finding the malware [10] and arrange it.

ARChon, Bliss, Bluestacksis used to learn about the data in the Android samples. It is capable of breaking down the Android files to extract data like XML and DEX files. The data are obtained from the malicious applications and built as an accentuate classification model. The requested permission is obtained from Manifest.xml. The application can perform a task only if it had been permitted by the user. Like for the android command READ CONTACTS, the application will gain permission to get the entire contact information. So we need to make sure that the permission given by the user depends upon the current task of the running application. The extracted application component which gives a description about the Activity, service and Broadcast receiver of the application. Since the malwares files carry the name of the original component, the component names occurring as notification are the families of the malware. The filters are obtained from the Manifest.xml. They are used to enhance interactions between android's internal components. Like for instance, when the system booting is finished, the application can start as soon as the device starts to run. The feature hardware is received from the Manifest.xml [11]. It provides permission to access the hardware. Like, if we want to access a camera, classes.dex issues the API command. A search is conducted in the broken codes of the application for checking if the command is permitted or not. So if any application tries to access a command that's not in their basic code, they can be determined as malicious [12]. For instance an application that has permission to send messages won't get permission each time it sends. So by receiving the data from classes.dex and analyzing it to find the original intention of the application may tell whether the application is original or malicious. Url, IP addresses, servers should also be taken into account along with user permissions to find an malware. They are then changed into vector space and data is fed to multi-level CNN or MLCNN. These seven attributes are the important parts of feature set. One of the most important concepts we have been seeing in the above methods is converting the data sets obtained into vector space to feed them into the deep learning [13] algorithm. During training the data's are needed for prediction most of the time. So these data are converted into numerical vectors. For converting, initially these seven attributes are combined into a single set L

$$L = L1 \cup L2 \cup \ldots \cup L7$$

If L is the set then the vector space dimension of the set is represented as |L|. The dimension is either 0 or 1. If the application is i then the vector is vi. It is described as

$$Wm \rightarrow ((m, l))$$

Where: $l \in L$ and $(m, l) = \{1$ *if application* m *contains features*, *otherwise* it is $0\}$ in order to solve the memory problem that arises due to high level of feature or data set that contains number of elements but the value of most of the elements is 0, the data set is converted into a sparse matrix. The vector Wm is converted. The value 0 is neglected in this method. The value 1 is alone included. Here label is either 0 or 10 if benign and 1 for malware. Feature - the index number of the application. The value is always 1.

5 Evaluation

Here we use two methods to evaluate our model that finds the malware applications [14]. To perform the evaluation a computer was used. In the first test, the evaluation performance of our model is analyzed [15] by cross checking it with the evaluation performance of four other machine learning models. In the second test, the performance while running is evaluated and cross checked with two other machine learning [16] models taken from the first test. For this nearly 2000 malicious and normal applications were taken by using an APK file crawler. The collected data were used for evaluations. Various formulas were used for calculating the data's measurement factors here:

TP (True positive) – are the perfectly arranged malware data's
FN (False negative) – wrongly arranged malware data's
TN (True Negatives) – perfectly arranged benign data's
FP (False Positive) – wrongly arranged benign data's.
Accuracy – it determines how similar the calculated value is to the standard value [4, 5].

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

Precision – how much the measured values are similar to each other.

$$Precision = \frac{tp}{tp + fp}$$

Recall/sensitivity – fraction of positive instances found to the total positive instances.

$$Recall = \frac{tp}{tp + fn}$$

Each researcher chooses their own method and data to perform detection and classification. A well knowledge person would definitely choose data that provide good information on the malicious application [17] to get accurate results. Because choosing poor data may lead to poor results. The MLCNN is initially tested for perfect effectiveness using a variety of feature sets. The combination of all the feature set gives a perfect value compared to others. But still the API [18] and string also provide a better value compared to the others. The permission in Table 2: Comparing it with machine learning methods, is a bit better than the intent feature set. But there might be many variables for these results. Like if the permission set contains only 200 features which are low for MLCNN. After this MLCNN is compared with three models that are based on machine learning (Linear SVM, SVMG–RBG, BPNN). The BPNN method uses the constant empirical value. The evaluation uses same data on both sides.

Table 2 shows that our MLCNN model is greater and accurate than the other machine learning models. To represent Fig. 1: The comparison between machine learning methods, about 98.5% of the malicious applications are detected which is greater than the other machine models. The proposed model is greater in precision, recall, accuracy, and F-score. Our model also produces fewer false alarms compared to the other models. Even though the other models are effective and the difference is smaller for some, our model is better than everything else.

Table 2. Comparing it with machine learning methods

Algorithm	Feature set			
	Precision	Recall	Accuracy	F-score
SVM	0.854	0.991	0.910	0.918
Linear SVM	0.910	0.990	0.920	0.925
SVMG–RBG	0.948	0.989	0.958	0.959
BPNN	0.955	0.988	0.980	0.981
MLCNN	0.975	0.988	0.985	0.985

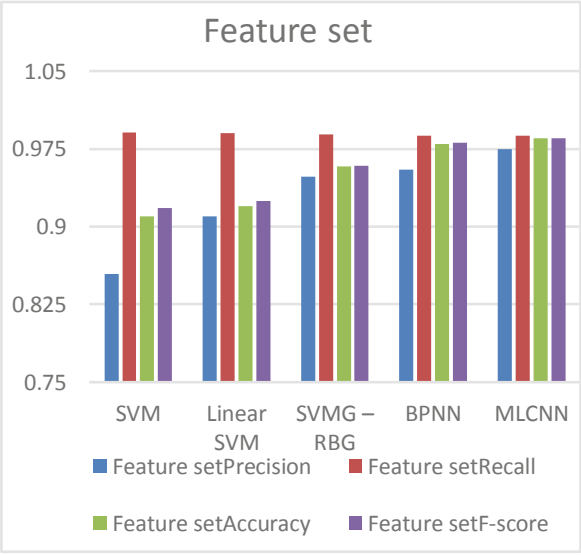


Fig. 1. The comparison between machine learning methods

6 Conclusion

The android malware detection model we proposed based on Convolutional neural Networks has provided MLCNN model which is accurate and higher than the other machine learning models and comparatively faster. It uses static, dynamic and hybrid analysis techniques to provide feature sets based on which the evaluation is done. The result is analyzed with the results of other models and it has proved that MLCNN can provide 98.5% accuracy and low false alarms compared to other models. Though the other models are effective their performance is comparatively low with our model. For future reference, our model can be subjected to dynamic analysis for more accuracy and speed.

References

1. Zegzhda, P., Zegzhda, D., Pavlenko, E., Ignatev, G.: Applying deep learning techniques for android malware detection. In: Proceedings of the 11th International Conference on Security of Information and Networks - SIN 2018 (2018). https://doi.org/10.1145/3264437.3264476
2. Backes, M., Nauman, M.: LUNA: quantifying and leveraging uncertainty in android malware analysis through Bayesian machine learning. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P) (2017). https://doi.org/10.1109/eurosp.2017.24
3. Peiravian, N., Zhu, X.: Machine learning for android malware detection using permission and API calls. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (2013). https://doi.org/10.1109/ictai.2013.53
4. Wu, W.-C., Hung, S.-H.: DroidDolphin: a dynamic android malware detection framework using big data and machine learning. In: RACS 2014, 5–8 October 2014, Towson, MD, USA (2014)
5. McLaughlin, N., Doupé, A., JoonAhn, G., Martinez del Rincon, J., Kang, B., Yerima, S., Zhao, Z.: Deep android malware detection. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy - CODASPY 2017 (2017). https://doi.org/10.1145/3029806.3029823
6. Zhang, Y., Yang, Y., Wang, X.: A novel android malware detection approach based on convolutional neural network. In: Proceedings of the 2nd International Conference on Cryptography, Security and Privacy - ICCSP 2018 (2018). https://doi.org/10.1145/3199478.3199492
7. Yeh, C.-W., Yeh, W.-T., Hung, S.-H., Lin, C.-T.: Flattened data in convolutional neural networks. In: Proceedings of the International Conference on Research in Adaptive and Convergent Systems - RACS 16 (2016). https://doi.org/10.1145/2987386.2987406
8. Canfora, G., Medvet, E., Mercaldo, F., Visaggio, C.A.: Acquiring and analyzing app metrics for effective mobile malware detection. In: Proceedings of the 2016 ACM on International Workshop on Security and Privacy Analytics - IWSPA 2016 (2016). https://doi.org/10.1145/2875475.2875481
9. Pang, Y., Xue, X., Wang, H.: Predicting vulnerable software components through deep neural network. In: Proceedings of the 2017 International Conference on Deep Learning Technologies - ICDLT 2017 (2017). https://doi.org/10.1145/3094243.3094245
10. Bhandari, S., Gupta, R., Laxmi, V., Gaur, M.S., Zemmari, A., Anikeev, M.: DRACO. DRACO: DRoid analyst combo an android malware analysis framework. In: Proceedings of the 8th International Conference on Security of Information and Networks - SIN 2015 (2015). https://doi.org/10.1145/2799979.280000
11. Lee, Y., Lee, J., Soh, W.: Trend of malware detection using deep learning. In: Proceedings of the 2nd International Conference on Education and Multimedia Technology - ICEMT 2018 (2018). https://doi.org/10.1145/3206129.3239430
12. Hou, S., Saas, A., Chen, L., Ye, Y., Bourlai, T.: Deep neural networks for automatic android malware detection. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM 2017 (2017). https://doi.org/10.1145/3110025.3116211
13. Kumar, R., Xiaosong, Z., Khan, R.U., Kumar, J., Ahad, I.: Effective and explainable detection of android malware based on machine learning algorithms. In: Proceedings of the 2018 International Conference on Computing and Artificial Intelligence - ICCAI 2018 (2018). https://doi.org/10.1145/3194452.3194465
14. Gonzalez, H., Kadir, A.A., Stakhanova, N., Alzahrani, A.J., Ghorbani, A.A.: Exploring reverse engineering symptoms in Android apps. In: Proceedings of the Eighth European Workshop on System Security - EuroSec 2015 (2015). https://doi.org/10.1145/2751323.2751330

15. Diao, W., Liu, X., Li, Z., Zhang, K.: Evading Android runtime analysis through detecting programmed interactions. In: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks - WiSec 2016 (2016). https://doi.org/10.1145/2939918.2939926
16. Ali Alatwi, H., Oh, T., Fokoue, E., Stackpole, B.: Android malware detection using category-based machine learning classifiers. In: Proceedings of the 17th Annual Conference on Information Technology Education - SIGITE 2016 (2016). https://doi.org/10.1145/2978192.2978218
17. Cakir, B., Dogdu, E.: Malware classification using deep learning methods. In: Proceedings of the ACMSE 2018 Conference on - ACMSE 2018 (2018). https://doi.org/10.1145/3190645.3190692
18. Kalgutkar, V., Stakhanova, N., Cook, P., Matyukhina, A.: Android authorship attribution through string analysis. In: Proceedings of the 13th International Conference on Availability, Reliability and Security - ARES 2018 (2018). https://doi.org/10.1145/3230833.3230849

Spatial Constrained K-Means for Image Segmentation

Yajuan Li[1], Yue Liu[1(✉)], Bingde Cui[1], Chao Sun[1], Xiaoxuan Ji[1], Jing Zhang[2], Bufang Li[3], Huanhuan Chen[1], Jianwu Zhang[1], Yalei Wang[1], and Xiaolin Wang[1]

[1] Hebei University of Water Resources and Electric Engineering, Handan, China
liuyue76@tju.edu.cn
[2] Tianjin Electronic Information College, Tianjin, China
[3] Cangzhou municipal human resources and Social Security Bureau, Cangzhou, China

Abstract. In this paper, we propose a novel spatial constrained clustering method, it is simple yet very effective, which has been validated it for image segmentation. The key observation of our model is that traditional K-Means clustering is popular for segmentation but it lacks effective spatial constraint. To address this issue, a general spatial constrained K-Means clustering framework is proposed and shows its effectiveness in image segmentation. Spatial constraints are expressed by points on adjacent positions, which cannot be segmented only by color gaps. With the expectation maximization algorithm, our method could be efficiently optimized, and a locally optimal solution could be guaranteed. Experiments on the Berkeley image segmentation dataset show that our method outperforms compared methods.

Keywords: K-Means · Spatial constraint · Image segmentation

1 Introduction

Image segmentation is one of the most basic and important fields in image processing and computer vision. It is the fundation for visual analysis and pattern recognition of images [1–3]. Image segmentation refers that an image is divided into non overlapping regions according to the features, typically including: gray scale, color, texture and shape feature, where these characteristics in the same area show the similarity and show obvious difference in different regions [8].

Clustering is a basic and important technique for exploratory data analysis [9–11], and has been widely applied to image segmentation [13–15,30]. There are many articles summarizing the use of clustering for image segmentation [5–7]. The clustering algorithms for image segmentation generally consider each pixel in the image as one data point and then perform clustering. Afterwards, the segmentation result [12,16,29] is obtained according to the clustering result. Among these clustering methods, K-Means algorithm is widely used due to its

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 662–672, 2021.
https://doi.org/10.1007/978-3-030-55180-3_50

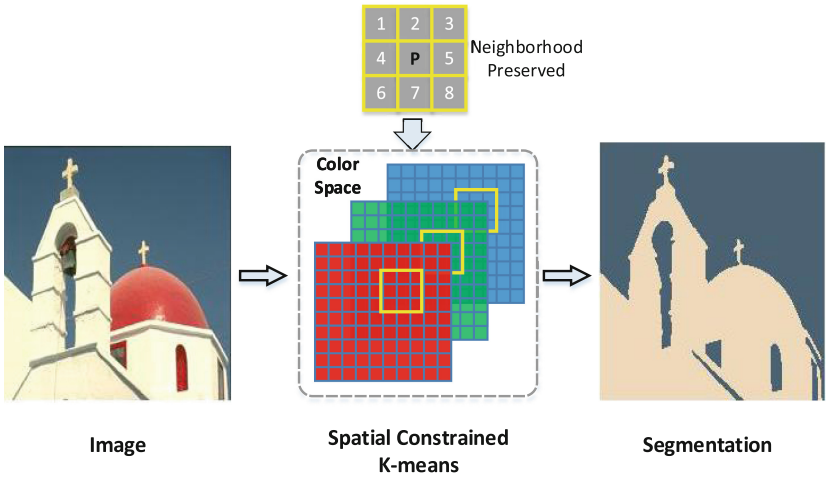


Fig. 1. The yellow boxes indicates the 8-neighborhood pixels of P, and in the process of image segmentation using K-Means algorithm, the spatial constraint is seamlessly incorporated.

simplicity and effectiveness. Although simple and interpretable, directly utilizing it for image segmentation is usually improper for the following reason. For image segmentation task, beyond the color attribute, the locations of pixels also should be taken into consideration for meaningful results. However, the result obtained by conventional clustering algorithm is usually not promising due to only considering the relationships in color space, i.e., the pixels with the similar colors are partitioned into one region, while the pixels with unsimilar colors are partitioned into different regions. Accordingly, the image segmentation by directly using K-Means algorithm is risky and unpromising [17]. To incorporate the spatial information into clustering, we introduce a constrained K-Means clustering algorithm which could be applied for image segmentation with both color feature and spatial information used.

To consider the constraint relationship between adjacent pixel points, we introduce the concept of pixel neighborhood. Our model is shown in Fig. 1, the category of this pixel is no longer dependent on the effect of the same color region and is related to the 8-pixel neighborhood of the point. As a result, the problem of only considering the color property is solved, and the real spatial constraint in the picture is added, and the segmentation effect is obviously improved.

To summarize, the main contribution of this work includes:

1. We propose a general constrained K-Means clustering algorithm, which can well address the image segmentation task by simultaneously making use of color feature and spatial information.
2. Our method could be effectively solved with Expectation Maximization (EM) algorithm, which makes our method not only effective but also efficient.

3. Experimental results on benchmark datasets demonstrate the advantages of our method.

2 Related Work

Threshold-based segmentation is a traditional image segmentation way. Due to its simple in implementation, small amount of computation and stable performance, it has become the most basic and widely applied segmentation technology in image segmentation. The classical Otsu algorithm [18] adaptively selects threshold based on the maximum inter class variance of the characteristics of the image itself. The edge detection-based method tries to solve the segmentation problem by detecting the edges of different regions. These methods are usually based on the observation that the gray values on the edge will show a significant change. The differential operator can be used for edge detection, such as Laplacian of a Gaussian (LoG)[1].

The region-based segmentation methods usually divide an image into different regions according to the similarity criterion. The basic idea of a region splitting and merging method [19] is to divide the image into a number of intersecting regions, and then these regions are divided or merged according to the relevant rules to complete the segmentation task, which are not only suitable for gray image segmentation, but also suitable for texture image segmentation. The representative method, watershed segmentation [20], is a mathematical morphology method based on topology theory.

The graph-based segmentation methods are associated with the problem of minimum cut problem of a graph, in which an image is usually mapped into a weighted undirected graph. Each node in the graph corresponds to a pixel in the image, with each edge connecting a pair of adjacent pixels. The representative methods include spectral clustering [21–23], affinity propagation [24] and incremental aggregation [25]. The K-Means algorithm is usually directly utilized for image segmentation [16,26–28], while they neglect the spatial information in the clustering process.

3 Our Method

3.1 K-Means Clustering with Image Segmentation

K-Means is one of the representative clustering algorithms. Suppose we are given a data set $\mathscr{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ with each data points $\mathbf{x}_n \in \Re^d$, we aim to partition this data set into K disjoint clusters $\mathscr{C}_1, \mathscr{C}_2, ..., \mathscr{C}_K$. Without considering the background of the problem, from the point of view of Euclidean space, we should gather the nearest point of the distance into a cluster, and the distance

[1] http://www.cse.dmu.ac.uk/sexton/WWWPages/HIPR/html/log.html.

between the points of different clusters is far away. The K-Means clustering objective is given by

$$\begin{aligned} &\min_{\gamma_{nk},\boldsymbol{\mu}_k} \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}||\mathbf{x}_n-\boldsymbol{\mu}_k||^2 \\ &s.t.\ \gamma_{nk}\in\{0,1\},\ \sum_{k=1}^{K}\gamma_{nk}=1, \end{aligned} \tag{1}$$

where $\gamma_{nk} \in \{0, 1\}$ indicates the assignment of the n^{th} point to the k^{th} cluster, and $\boldsymbol{\mu}_k$ is the k^{th} cluster centroid.

Specifically, each pixel in a color image is considered as a data point in a 3-dimensional space that corresponds to the intensity of RGB. Then each pixel is replaced by its corresponding cluster center, and the image is reconstructed. The traditional K-Means clustering only considers the distance in color space while lacks effective spatial constraint.

3.2 Our Objective Function

According to the analysis above, we combine the requirements of the spatial constraint of the 8-neighborhood pixels with the K-Means algorithm to obtain our new objective function as follows:

$$\begin{aligned} &\min_{\gamma_{nk},\mu_k} \sum_{n=1}^{N}\sum_{k=1}^{K}\left(\gamma_{nk}\|\mathbf{x}_n-\boldsymbol{\mu}_k\|^2+\alpha\sum_{p=1}^{P}|\gamma_{nk}-\gamma_{pk}|\right) \\ &s.t.\ \ \gamma_{nk}\in\{0,1\},\ \sum_{k=1}^{K}\gamma_{nk}=1, \end{aligned} \tag{2}$$

where $\alpha > 0$ is a hyperparameter to balance the reconstruction error and the violation of spatial constraint. $\gamma_{nk} \in \{0, 1\}$ indicates the assignment of the n^{th} point to the k^{th} cluster, and $P \in \{3, 5, 8\}$ indicates the number of neighborhood points, as shown in Fig. 2. Specifically, the numbers of neighborhood pixels are 3, 5, and 8 for the pixel located on the corner, boundary and interior, respectively. Adding spatial constraint items makes the clustering effect not only affected by the similar colors, but pixels that are in close proximity are more likely to be divided into the same category.

We optimize our objective function in Eq. (2) through the EM algorithm, and alternatively optimize the objective function with respect to γ_{nk} and $\boldsymbol{\mu}_k$. Specifically, by taking the derivative of the objective function with respect to $\boldsymbol{\mu}_k$, we have

$$-2\sum_{n=1}^{N}\gamma_{nk}(\mathbf{x}_n-\boldsymbol{\mu}_k)=0. \tag{3}$$

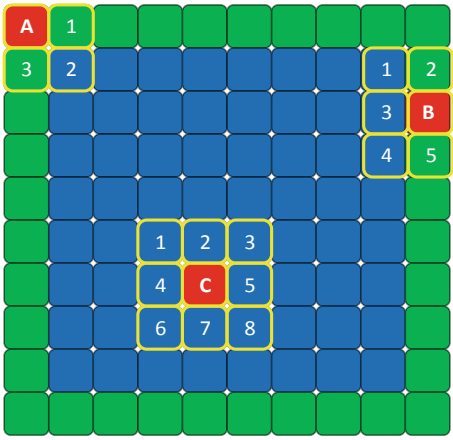


Fig. 2. The yellow box represents the 8-pixel neighborhood of points, the figure shows all the cases of the 8-pixel neighborhood in the image.

Accordingly, we can update $\boldsymbol{\mu}_k$ with the following rule:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} \gamma_{nk}}. \tag{4}$$

Similar to K-Means clustering, the assignments can be updated with the following rule

$$\gamma_{nk} = \begin{cases} 1, & \text{if } k = \arg\min_k ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2 + \alpha \sum_{p=1}^{P} |\gamma_{nk} - \gamma_{pk}| \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

According to the above formulation, we can see that our method is relatively simple in form and to optimize. The solution is basically consistent with the ordinary K-Means algorithm, but ours considers the effect of neighborhood constraints, which makes the image segmentation more reasonable. For clarity, the proposed method is summarized in Algorithm 1.

4 Experiments

In the experiments, we conduct our method for image segmentation task, and compared ours with existing state-of-the-art image segmentation methods.

Algorithm 1. Spatial K-Means for Image Segmentation

Input: An original image, and K.

1: Convert the image to the RGB feature matrix and randomly initialize each point's γ_{nk} according to the K value.
2: $n = 0$
3: **repeat**
4: $n = n + 1$
5: Update $\boldsymbol{\mu}_k$ according to Eq. (4)
6: Update γ_{nk} according to Eq. (5)
7: Update the objective function according to Eq. (2)
8: **until** the objective function converges.

Output: An image of the result of the image segmentation.

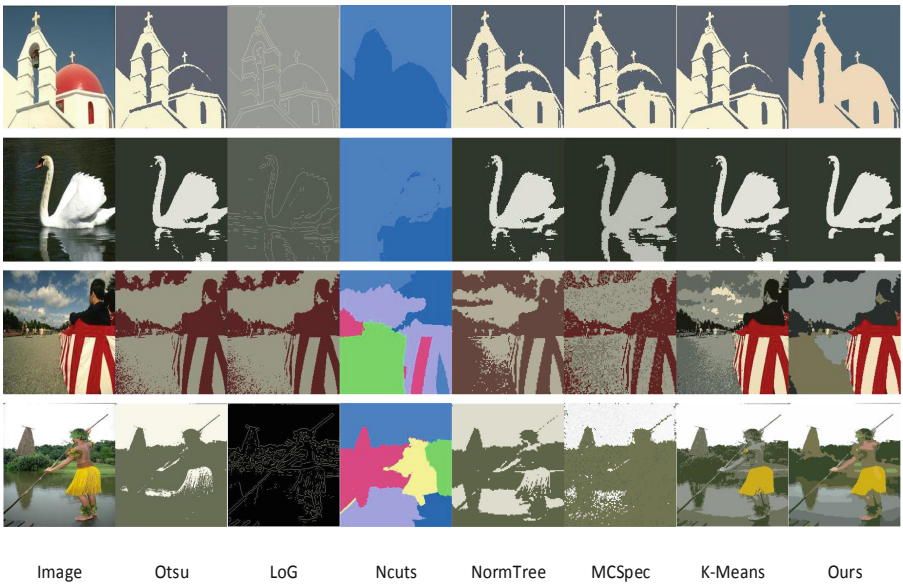


Fig. 3. Image segmentation results compared with the classical Otsu algorithm (Otsu) [18], the Laplacian of a Gaussian (LoG), the normalized cut algorithm (NCuts) [21], the normalized partitioning tree (NormTree) [22], the multiclass spectral (MCSpec) [23], and K-Means.

4.1 Experiment Setting

Dataset. We test our method on the well-known Berkeley image segmentation dataset [31] - Berkeley Segmentation Data Set (BSDS500)[2] - consisting of 500 natural images, ground-truth human annotations, and benchmark codes.

Evaluation Metrics. To quantitatively evaluate the overall segmentation performance, we employ five different quality metrics: Accuracy (ACC), Rand Index

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/.

Table 1. Performance (mean ± standard deviation) of comparisons. Comparison of different methods on Berkeley image segmentation dataset by Accuracy (ACC), Rand Index (RI), Precision, Recall and F1-measure. The best results are highlighted in bold. Our method shows competetive results compared to state-of-the-arts.

Methods	Metrics(%)				
	ACC	RI	Precision	Recall	F1-measure
Otsu	56.23 ± 12.21	61.60 ± 7.96	47.05 ± 15.83	80.40 ± 15.53	56.57 ± 12.25
LoG	47.69 ± 16.44	42.45 ± 10.27	36.39 ± 17.17	$\mathbf{82.19 \pm 4.52}$	48.48 ± 15.39
NCuts	53.63 ± 11.61	72.82 ± 14.18	73.03 ± 11.06	49.59 ± 20.18	50.98 ± 11.46
NormTree	59.86 ± 12.83	61.71 ± 8.53	46.29 ± 16.43	72.80 ± 17.06	57.19 ± 14.11
MCSpec	43.39 ± 11.20	56.64 ± 12.84	42.03 ± 17.05	59.72 ± 24.93	43.88 ± 12.03
K-Means	46.77 ± 8.45	72.20 ± 12.53	$\mathbf{73.32 \pm 12.04}$	42.42 ± 7.67	48.25 ± 8.67
Ours	$\mathbf{62.36 \pm 10.61}$	$\mathbf{74.96 \pm 11.42}$	62.68 ± 11.79	75.91 ± 12.45	$\mathbf{64.56 \pm 11.04}$

(RI) [32], Precision, Recall, and F1-measure [4]. For all these metrics, large value indicates good segmentation result. For accuracy, we specify the definition used in our experiments as follows: given a data point $\mathbf{x}_i$, we denote the result label and ground-truth label as ω_i and c_i, respectively, then we have

$$ACC = \frac{\sum_{n=1}^{N} \delta(c_i, map(\omega_i))}{N}, \tag{6}$$

where $\delta(x, y) = 1$ when $x = y$, otherwise $\delta(x, y) = 0$. $map(\omega_i)$ is the permutation map function, which maps the result labels into ground-truth labels. The best map can be obtained by Kuhn-Munkres algorithm.

4.2 Experiment Results

We compared our method with the existing state-of-the-art segmentation algorithms: the classical Otsu algorithm (Otsu) [18], the Laplacian of a Gaussian (LoG), the normalized cut algorithm (NCuts) [21], the normalized partitioning tree (NormTree) [22], the multiclass spectral (MCSpec) [23], and K-Means. We use the Berkeley Segmentation Data Set and repeated 10 times for each image. Table 1 summarizes the performance for all methods. Our proposed method shows competitive results compared with state-of-the-art methods. It is observed that although the performance of our method are not the best in terms of Precision and Recall, they are also comparable. More importantly, the results in terms of F1-measure which is more comprehensive than Precision and Recall, further verify the advantage of our method over all the other compared approaches. The scores of ACC and RI also validate that our method is much better than the other methods. For visual comparison, we demonstrate the segmentation results in Fig. 3. We can see that the segmentation results of ours are more accurate than other state-of-the-art segmentation algorithms. Accordingly, Table 1 and Fig. 3 demonstrate the advantages of our method, both quantitatively and qualitatively.

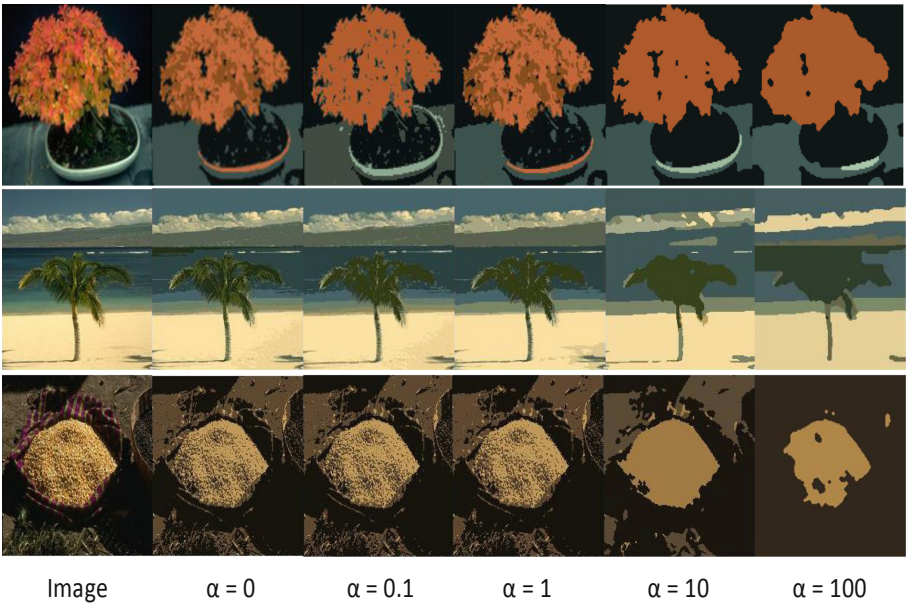


Fig. 4. $\alpha > 0$ is hyperparameter to balances the pixels distance constraint error and the pixels position constraint error for the whole objective function. As shown in the figure, when $\alpha = 0$ the effect is the result of the common K-Means algorithm. With the increase of α, the constraint ability of the 8-pixel neighborhood can be clearly seen, which makes the image segmentation effect obvious.

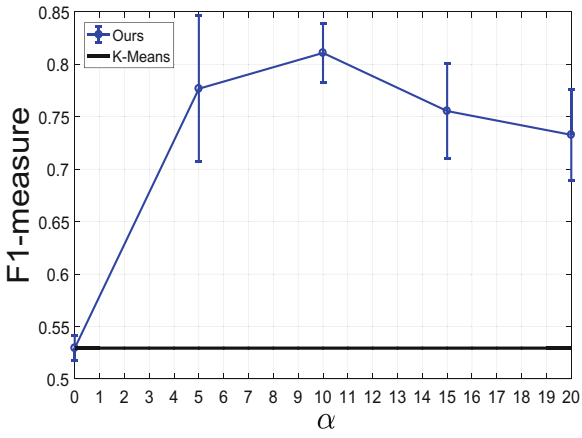


Fig. 5. The effect of the value of α on the evaluation metrics. Set α from 0 to 20, the blue line is the average and standard deviation of the performance curve of 5 random run. We observed the F1-measure, and our method is always better than K-Means, so we set $\alpha = 10$ all the time in our experiment.

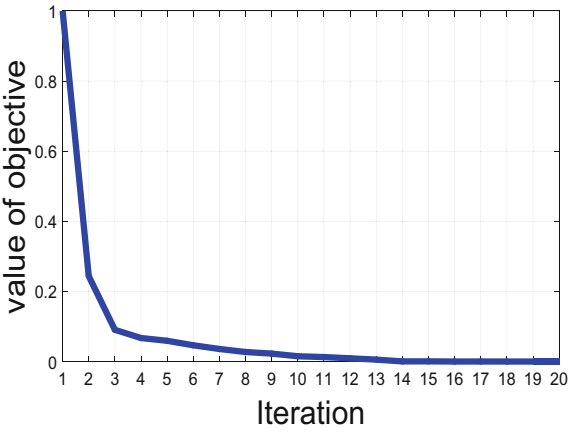


Fig. 6. Convergence curves on Berkeley Segmentation Data Set (BSDS500). It is clear in the figure that several iterations are required to quickly approach convergence (normalized to $[0, 1]$).

We also provide the investigation for the hyperparameter α. Specifically, with the same image as input, we vary the value of α and report image segmentation results. As shown in Fig. 4, the importance of spatial information is clearly shown. For $\alpha = 0$, our method is actually degraded to the conventional K-Means. With the value of α getting larger, we can find that the segmentation results is consistent with the assumption of our model. Furthermore as shown in Fig. 5, by continuously varying the value of α, the F1-measure scores are reported. We find that our method is always better than the K-Means method since with a non-zero value of α the performance is much better than that of K-means (when $\alpha = 0$). For simplicity, we set $\alpha = 10$ in our experiments.

As shown in Fig. 6, the value of our objective function decreases monotonically with the iterative optimization, which is consistent with the theoretical analysis. For conciseness, the axes are normalized to $[0, 1]$. It is worth noting that, although the computational cost is high for image data (e.g. an image of 100×100 is equivalent to 10,000 sample points), our model holds the flexibility to balance the performance and computation time due to the fast convergence rate at the beginning of a small number of iterations.

5 Conclusion

In this work, we propose a novel and simple image segmentation method by extend K-Means clustering into spatial constrained clustering. We take the constraints of the natural spatial information in image into consideration, and obtain promising segmentation result. Although plenty of the existing literatures also use K-means algorithm for image segmentation, they usually directly applied

K-Means for segmentation task. Instead, our work incorporate the spatial information into K-Means as a unified framework. In future, we will extend our model for hierarchical clustering and more real-world applications will be considered.

Acknowledgments. This work was supported in part by Hebei provincial science and technology plan self-raised project (Grand No: 18212108), Cangzhou science and technology research and development project (Grand No: 172104002), Hebei University Science and technology research project (Grand No: QN2019191) and Social science development research project of Hebei Province (Grand No: 2019030101017).

References

1. Cheng, M.M., Liu, Y., Hou, Q., et al.: HFS: hierarchical feature selection for efficient image segmentation. In: Computer Vision C ECCV 2016. Springer (2016)
2. Wang, Z., Feng, J., Yan, S., et al.: Image classification via object-aware holistic superpixel selection. IEEE Trans. Image Process. A Publ. IEEE Signal Process. Soc. **22**(11), 4341–52 (2013)
3. Wang, S., Lu, H,. Yang, F., et al.: Superpixel tracking. In: International Conference on Computer Vision. IEEE Computer Society, pp. 1323–1330 (2011)
4. Powers, D.M.W.: Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol. **2**, 2229–3981 (2011)
5. Yuheng, S., Hao, Y.: Image segmentation algorithms overview. arXiv preprint arXiv:1707.02051 (2017)
6. Dhanachandra, N., Chanu, Y.J.: A survey on image segmentation methods using clustering techniques. Eur. J. Eng. Res. Sci. **2**(1), 15–20 (2017)
7. Dhanachandra, N., Chanu, Y.J.: A new approach of image segmentation method using K-means and kernel based subtractive clustering methods. Int. J. Appl. Eng. Res. **12**(20), 10458–10464 (2017)
8. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recogn. **38**(9), 1277–1294 (1993)
9. Ackerman, M., Ben-David, S., Branzei, S., et al.: Weighted clustering. In: AAAI Conference on Artificial Intelligence (2011)
10. Hartigan, J.A., Wong, M.A.: A K-means clustering algorithm. Appl. Stat. **28**(1), 100–108 (1979)
11. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: International Conference on Neural Information Processing Systems: Natural and Synthetic, pp. 849–856. MIT Press (2001)
12. Ng, H.P., Ong, S.H., Foong, K.W.C., et al.: Medical image segmentation using k-means clustering and improved watershed algorithm. In: IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 61–65. IEEE Computer Society (2006)
13. Zhou, D., Huang, J.: Learning with hypergraphs: clustering, classification, and embedding. In: International Conference on Neural Information Processing Systems, pp. 1601–1608. MIT Press (2006)
14. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
15. Agarwal, S., Lim, J., Zelnik-Manor, L., et al.: Beyond pairwise clustering. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 838–845. IEEE (2005)

16. Chen, T.W., Chen, Y.L., Chien, S.Y.: Fast image segmentation based on K-Means clustering with histograms in HSV color space. In: IEEE Workshop on Multimedia Signal Processing, pp. 322–325. IEEE (2008)
17. Guo, Y., Xia, R., et al.: A novel image segmentation approach based on neutrosophic C-means clustering and indeterminacy filtering. Neural Comput. Appl. **28**(10), 3009–3019 (2017)
18. Ohtsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (2007)
19. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2 edn. (2002)
20. Meyer, F.: The watershed concept and its use in segmentation: a brief history. arXiv preprint arXiv:1202.0216 (2012)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
22. Wang, J., Jia, Y., Hua, X..S, et al.: Normalized tree partitioning for image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
23. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: International Conference on Computer Vision, pp. 313–319 (2003)
24. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007)
25. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vision **59**(2), 167–181 (2004)
26. Pappas, T.N.: An adaptive clustering algorithm for image segmentation. IEEE Trans. Signal Process. **40**(4), 901–914 (1992)
27. Zhao, Y., Gao, Z., Mi, B.: Fast image segmentation based on k-means algorithm. In: Proceedings of the 4th International Conference on Internet Multimedia Computing and Service, pp. 166–169. ACM (2012)
28. Luo, M., Ma, Y.F., Zhang, H.J.: A spatial constrained k-means approach to image segmentation. In: Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia, vol. 2, pp. 738–742. IEEE (2003)
29. Dhanachandra, N., Manglem, K., Chanu, Y.J.: Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Proc. Comput. Sci. **2015**(54), 764–771 (2015)
30. Wang, L., Pan, C.: Robust level set image segmentation via a local correntropy-based K-means clustering. Pattern Recogn. **47**(5), 1917–1925 (2014)
31. Martin, D., Fowlkes, C., Tal, D., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 2, pp. 416–423. IEEE (2002)
32. Rand, W.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)

On-Line Recognition of Fragments of Standard Images Distorted by Non-linear Devices and with a Presence of an Additive Impulse Interference

Nataliya Kalashnykova[1], Viktor V. Avramenko[2], Viacheslav Kalashnikov[2,3,4], and Volodymyr Demianenko[2(✉)]

[1] Universidad Autónoma de Nuevo León (UANL), Ave. Universidad S/N, 66455 San Nicolás de los Garza, NL, Mexico
[2] Sumy State University (SumDU), Rimsky-Korsakov st. 2, Sumy 40007, Ukraine
vldemyan@gmail.com
[3] Tecnologico de Monterrey (ITESM), Ave. Eugenio Garza Sada 2501 Sur, 64849 Monterrey, NL, Mexico
[4] Central Economics and Mathematics Institute (CEMI), Nakhimovsky pr. 47, Moscow 117418, Russia

Abstract. On the base of the first-order disproportion functions, an algorithm recognizing fragments of standard images under conditions when the analyzed signal contains these fragments in a distorted form due to passing through a non-linear device, the static characteristic of which can be represented by a polynomial with unknown coefficients, is developed. Both, continuous signals and those, described by discrete pixel brightness values of a video image, are considered with the presence of additive impulse noises.

Keywords: Image recognition · Standard fragments · Disproportion functions · Additive impulse interferences · Signal distortion · Integral disproportionality functions · Static characteristic · Nonlinear device · On-line recognition · Multiplicative interference

1 Introduction

There is a wide class of problems, to solve which, it is necessary to recognize standard signals. For example, during flaw detection, it is necessary to recognize oscillograms characteristic of a certain defects type [1].

The problem of recognition of acoustic pulsed signals against a background of industrial noise arises during coal mining [2].

The need for automatic identification of one of the predefined sound samples exists during radio signals transmission [3].

Waveform recognition is also used in Asynchronous Address Communication Systems. Each channel (subscriber) is assigned to a specific waveform, which is the hallmark of this subscriber [4].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 673–685, 2021.
https://doi.org/10.1007/978-3-030-55180-3_51

Also, the need to recognize fragments of standard images arises in the processing of images received in real time from camcorders.

In practice, the recognition problem has to be solved in the presence of a number of factors which complicate problem solution significantly. First of all, recognition should occur promptly based on current data. In addition, not the entire standard object can arrive for recognition, but only its fragment. It should also be borne in mind that, as a rule, the amplitude of the signal depends on its reception and transmission conditions. Therefore, its value can be arbitrary. For example, the signal from the camcorder depends on the object brightness. In wireless data transmission, the signal level depends on the distance, direction of the transmitting and receiving antennas, and many other factors.

In addition, often when transmitting a signal over a communication channel, it undergoes non-linear distortion. These distortions appear when a signal passes through a device which static characteristic has non-linear sections. For example, such device as video amplifier. Its static characteristic is nonlinear near zero [5].

In the general case, the input signal can be so weak that its amplitude does not exceed the width of the nonlinear part of the static characteristic of the video amplifier. Therefore, its recognition becomes almost impossible due to unknown values of the nonlinearity parameters. In addition, these parameters can randomly change over the time.

It should also be borne in mind, that interferences that may be superimposed on the transmitted signal, usually, have unknown characteristics.

Thus, it is required to carry out recognition of fragments of standard signals from current data in the presence of interference and non-linear distortions of the analyzed signal. The recognition should be invariant with respect to the amplitude of the standard signal.

There are many different ways to recognize signals. For the recognition of affine-distorted images there is an effective method of normalization [6].

A non-classical approach to solve the problem of signal recognition in automated radio monitoring is described in [7]. It is proposed to use the decision rule for selecting and recognizing these signals in the presence of unknown signals. This rule is based on a description of signals operating in the frequency channel using a probabilistic model in the form of orthogonal extensions.

Operational recognition of fragments and signal complexes and the allocation of video data objects using the object systems of wireless networks was developed in [8]. In this case, the so-called weighty values are used. Among them, with the help of information parameters, significant weighty values are selected. They are the basis for operational data processing.

The task of accelerating the process of detecting objects in the images was solved in [9]. Multiscale scanning is used. To solve this problem, it is proposed to use preliminary processing of candidates, using integral characteristics. This processing is implemented as the first stage of the cascade of classifiers of a mixed type.

The neural networks are used for pattern recognition [10, 11]. In particularly, deep neural networks-based (DNN) algorithms are quite productive and efficient for pattern recognition [12]. Nevertheless, one of the minor points of the DNN is the necessity to accumulate databases of enormous size to make the DNN methods useful. For example,

the ImageNet database is the outcome of a collaboration between Stanford University and Princeton University and has become a reference in the field of computer vision. It contains around fourteen million images originally labeled with Synsets of the WordNet lexicon tree.

Also, wavelet analysis [13] is used to solve the problem.

However, all these methods require either training or observation of the analyzed signal for a certain period of time.

In [14], operational recognition of fragments of standard signals in the presence of additive or multiplicative noise was considered. But, at the same time, the standard signals were not distorted. Also, the video signals recognition was not considered.

The principle of recognition of a distorted image is given in [15]. It describes methods for recognizing fragments of one-dimensional and two-dimensional both continuous and discrete standard signals. These signals are distorted due to passage through nonlinear devices. In addition, the recognition is carried out in the presence of random interference with unknown characteristics. Cases where the interference appears and disappears at random times, so that, they are pulsed by nature, are considered.

The standard signal is included to the analyzed signal with some constant, previously unknown scale factor. In addition, it is believed that the nonlinear part of the static characteristic of the device, through which the analyzed signal passes, can be represented by a polynomial with unknown coefficients.

According to the current data of the analyzed signal and the known data of the standards, it is necessary to quickly recognize moments, when the interference disappears and to recognize fragments of the standard signals during these intervals.

For 2-D images, they are assumed to be of the same size and without rotation. The listed conditions, in particular, correspond to stationary cameras that are installed under arbitrary low light conditions.

To solve the problem, disproportion functions are used [15, 16], the practical application of which can be found in more detail in [17–21].

2 Disproportionality Functions

The Disproportionality or shortly the disproportion functions are characteristics of numerical functions. They allow to quantify the deviation of the relationship between two numerical functions from the proportional relationship.

The following types of disproportionalities are distinguished [15]:

- disproportionality with respect to the derivative of n-order;
- successive n-order disproportionalities;
- disproportionality in the value of n-order;
- relative disproportionality of n-order;
- n-order integral disproportionality [17].

The following is a summary of the disproportion functions used in this paper.

If there is a proportional relationship, all these disproportionalities are equal to zero, regardless of the value of the coefficient of proportionality.

So the disproportion function with respect to the derivative of the n-th order of the function $y(x)$ with respect to x is described by the expression:

$$@d_x^{(n)}y = \frac{y}{x^n} - \frac{1}{n!} \cdot \frac{d^n y}{dx^n} \tag{1}$$

This disproportion is equal to zero for the power function $y = kx^n$ regardless of the factor k. Here $n \geq 1$ is an integer.

For $n = 1$:

$$@d_x^{(1)}y = \frac{y}{x} - \frac{dy}{dx} \tag{2}$$

The @ symbol is chosen to indicate the operation of disproportion calculating, d - from the English derivative. The left side is read "at d one y by x".

For the functions $y = \psi(t)$ and $x = \varphi(t)$ defined parametrically (t is a parameter), the disproportion with respect to the derivative of the n-th order (1) is calculated taking into account the rules of finding $d^n y/dx^n$ with the parametric dependence of y on x.

In particular, for $n = 1$

$$@d_x^{(1)}y = @d_{\varphi(t)}^{(1)}\psi(t) = \frac{y}{x} - \frac{y'_t}{x'_t} = \frac{\psi(t)}{\varphi(t)} - \frac{\psi'(t)}{\varphi'(t)}. \tag{3}$$

Obviously, if $\psi(t) = k\varphi(t)$, then disproportionality (3) is equal to zero in the entire area of existence $x = \varphi(t)$, regardless of the value of k.

It is easy to verify that disproportion functions have the following properties:

1. Multiplication of the function y by any scalar m leads to the multiplication of its disproportion function by the same scalar.
2. The disproportion function of the sum (difference) of numerical functions is equal to the sum (difference) of their disproportion functions.

Remark 1. In other words, the operator $@d_x^{(n)}$ defined on the space $C^n(\Omega)$ of n times continuously differentiable real functions is linear on this space.

To detect polynomial dependencies, the so-called sequential disproportion function with respect to the first order derivative is used [15].

This is a sequential calculation of disproportion for previously calculated disproportion. For example, disproportion (3) was first calculated with respect to the first-order derivative of the function $y(x)$ with respect to x.

For the result obtained, the disproportion (3) with respect to x is again calculated. So repeated S-times. The result is the so-called value. S-disproportion (SDF) with respect to the 1-st order derivative.

For example, the first-order S-disproportion function for $S = 3$ is obtained by the formula

$$@(3)d_{x(t)}^{(1)}y(t) := @d_{x(t)}^{(1)}\left\{@d_{x(t)}^{(1)}\left[@d_{x(t)}^{(1)}y(t)\right]\right\} \tag{4}$$

Its calculation is performed in steps, the number of which is equal to the degree of the polynomial. In the last step, the disproportion is zero, regardless of the values of the coefficients of the polynomial.

This can be seen in the example of calculating S-disproportionality (4) for a third-order polynomial $y = a_3x^3 + a_2x^2 + a_1x$

Step 1: $z_1 = @d_x^{(1)}y = \frac{y}{x} - y' = -2a_3x^2 - a_2x$
Step 2: $z_2 = @d_x^{(1)}z_1 = \frac{z_1}{x} - z_1' = 2a_3x$
Step 3: $z_3 = @d_x^{(1)}z_2 = \frac{z_2}{x} - z_2' = 0$

An equality SDF to zero is used in signal recognition.

3 Overview of Methods for Recognizing Undistorted Standard Signals with the Presence of Impulse Interference

A finite set of reference signals described by functions $f_i(t)$, where $t \in [0; T_i]$, $i = 1, 2, \ldots, M$.

These functions are smooth, continuous, having first derivatives.

In the presence of additive interference, the analyzed signal is described by the expression:

$$y(t) = kf_i(t + \tau_i) + \eta(t) \tag{5}$$

where $f_i(t)$is the i-th standard function;
$\tau_i \in [0; T_i]$ - time shift between the signal and the i-th standard;
$\eta(t)$ - additive interference, which is known only that it can disappear and appear at random points in time;
k - is a coefficient whose value is unknown

For multiplicative interference

$$y(t) = kf_i(t + \tau_i)\eta(t) \tag{6}$$

It is necessary to determine from the current values of the signal $y(t)$ and its first derivative which of the standard functions is present at the given moment in the analyzed signal.

For the case when the analyzed signal is described by expression (5) or (6), the disproportion function with respect to the first-order derivative for the numerical functions defined parametrically [15, 16] was used for this purpose [14]. This disproportion of function (5) with respect to the standard function $f_j\left(t + \tau_j\right)$ is described by the expression:

$$\begin{aligned} @d^{(1)}_{f_j(t+\tau_j)}y(t) &= \frac{kf_i(t + \tau_i) + \eta(t)}{f_j\left(t + \tau_j\right)} - \frac{kf_i^{'}(t + \tau_i) + \eta'(t)}{f_j^{'}\left(t + \tau_j\right)} \\ &= k@d^{(1)}_{f_j(t+\tau_j)}f_i(t + \tau_i) + @d^{(1)}_{f_j(t+\tau_j)}\eta(t) \end{aligned} \tag{7}$$

For the case when $j = i$, disproportion (7) has the form:

$$@d^{(1)}_{f_i(t+\tau_i)} y(t) = \frac{\eta(t)}{f_i(t+\tau_i)} - \frac{\eta'}{f_i'} = @d^{(1)}_{f_i(t+\tau_i)} \eta(t). \tag{8}$$

Obviously, when the interference disappears, $\eta(t) = 0$, $\eta'(t) = 0$, and with a correctly selected time shift $\tau_i \in [0; T_i]$, disproportion (8) is equal to zero. Thus, the fact that disproportion (7) is equal to zero indicates that at time t the interference disappeared, and the standard function $f_i(t)$ shifted by τ_i is present in the analyzed signal. For other reference functions, disproportion (7) will not be zero at any time shifts. An exception may be the case when several standards have matching fragments.

In the presence of multiplicative interference, when the analyzed signal is described by expression (6), the disproportion (3) of $y(t)$ with respect to $f_i(t+\tau)$ for time shift τ_i has the form:

$$@d^{(1)}_{f_i(t+\tau_i)} y(t) = -k\eta'(t)\frac{f_i(t+\tau_i)}{f_i'(t+\tau_i)} \tag{9}$$

At the moment when the derivative of the interference $\eta'(t) = 0$, disproportion (9) becomes equal to zero. That is, in this case, the operational recognition of the standard signal occurs even in the presence of interference.

4 Mathematical Formulation of the Problem for Nonlinear Distortion Without Interference

Consider the case when the standard signal $f_i(t + \tau_i)$ arrives at the non-linear area of the device's static characteristic, so it is distorted. Suppose that non-linearity can be represented by a p-th degree polynomial with zero free term. We also assume that $\tau_i = 0$.

Then the analyzed signal at the output of a nonlinear device is described by the expression:

$$y(t) = a_p f_i^p(t) + a_{p-1} f_i^{p-1}(t) + \ldots + a_1 f_i(t) \tag{10}$$

It is required to recognize the standard signal at the current time t with unknown coefficients of the polynomial (10). To do this, S-disproportion (SDF) (4) can be used. When it is calculated using the standard function $f_i(t)$, at the p-th step SDF (t) will be equal to zero.

Consider an example for the case when $p = 3$.

Step1. The disproportion $z_1(t)$ (3) of $y(t)$ with respect to $f_i(t)$ is calculated:

$$z_1(t) = @d^{(1)}_{f_i(t)} y(t) = \frac{y(t)}{f_i(t)} - \frac{y'(t)}{f_i'(t)} = -2a_3 f_i^2(t) - a_2 f_i(t) \tag{11}$$

Step 2. The disproportion $z_2(t)$ (3) of $z_1(t)$ with respect to $f_i(t)$ is calculated:

$$z_2(t) = @d^{(1)}_{f_i(t)} z_1(t) = \frac{z_1(t)}{f_i(t)} - \frac{z_1^{'}(t)}{f_i^{'}(t)} = 2a_3 f_i(t) \tag{12}$$

Step 3. The disproportion $z_3(t)$ (3) of $z_2(t)$ with respect to $f_i(t)$ is calculated:

$$SDF(t) = z_3(t) = @d^{(1)}_{f_i(t)} z_2(t) = \frac{z_2(t)}{f_i(t)} - \frac{z_2'(t)}{f_i'(t)} = 0 \tag{13}$$

The fact that $z_3(t)$ is equal to zero indicates that at the current time t, a fragment of the standard signal $f_i(t)$ is being input to the recognition system. For all other standard functions, this condition will not be met. Exceptions is possible if different standard functions contain the same fragment.

In the common case, a polynomial describing a nonlinear portion of the static characteristic of a device may contain a free term that is non-zero. To use the method proposed above, it is proposed to differentiate the analyzed signal $y(t)$ and use standard function derivatives instead of the standard functions themselves.

If the order of the polynomial is unknown, then it is proposed to gradually increase it to a predetermined $p_{\max}$ value. If at its achievement $SDF(t)$ does not equal to zero for any of the standards, we can assume that the recognition did not take place.

5 Recognition of Distorted Video Image Without Interference

First, we consider the problem when there are no interferences. For a digital camera, e.g., camcorders the images are represented as two-dimensional arrays of pixels. Each pixel has its own color value. These are intensity of red, green, and blue components. Each of them can vary from 0 to 255.

Assume that there are m reference images (standards) represented by matrices of pixels. Suppose also that after scanning we have the arrays of red $R_k = \{R_k(q)|1 \le q \le N\}$, green $G_k = \{G_k(q)|1 \le q \le N\}$, and blue $B_k = \{B_k(q)|1 \le q \le N\}$, brightness values for every pixel and for every standard, where the notation is explained in the table below:
Nomenclature:

$k = 1,2, ..., m$	is the order number of the standard;
$q = i \cdot w + j$	is the pixel order number on the screen;
w	is the number of pixels in one line; $w \ge 1$;
j	is the pixel order number in a row; $1 \le j \le w$;
h	is the total number of rows in the screen;
I	is the line number; $0 \le i \le h$;
$N = (1+h) \cdot w$	is the total number of the pixels on the screen.

Similarly, the measured (output) red, green and blue components of the color pixels of the analyzed image are arranged in the arrays $r_k = \{r_k(q)|1 \le q \le N\}$, $g_k = \{g_k(q)|1 \le q \le N\}$ and $b_k = \{b_k(q)|1 \le q \le N\}$ respectively ($k = 1, 2, \ldots, m$).

Assume, as in the previous case, that the nonlinear part of the static characteristic is described by a polynomial of degree p with zero free term.

If the pixel in the analyzed image corresponds to the pixel of the k-th standard image, their values of red, green and blue brightness, taking into account non-linear distortions, are described by polynomials with unknown coefficients. For example, for $p = 3$ they have the form:

$r_k(q) = a^r_{k3}R^3_k(q) + a^r_{k2}R^2_k(q) + a^r_{k1}R_k(q), 1 \leqslant q \leqslant N$
$g_k(q) = a^g_{k3}G^3_k(q) + a^g_{k2}G^2_k(q) + a^g_{k1}G_k(q), 1 \leqslant q \leqslant N$
$b_k(q) = a^b_{k3}B^3_k(q) + a^b_{k2}B^2_k(q) + a^b_{k1}B_k(q), 1 \leqslant q \leqslant N,$
where $a^r_{k3}, a^r_{k2}, a^r_{k1}, a^g_{k3}, a^g_{k2}, a^g_{k1}, a^b_{k3}, a^b_{k2}, a^b_{k1}, k = 1, K, m$ - unknown coefficients.

Thus, the task is to recognize the fragment of the standard image by the current values of the brightness of the pixels and the data of the standard images. It should be performed, despite the fact that the parameters of the nonlinear part of the static characteristic of the device through which the signal passes are unknown.

As in the previous case, you can use the sequential disproportion function (SDF) for this. However, it should be noted that the pixel luminance's of the standard signals are measured in integers from 0 to 255. Therefore, instead of the disproportion (3) used in the calculation of SDF (4) for continuous functions, the first-order integral disproportions can be used [17].

$$I(q) = @I^{(1)}_{R_k} r_k(q) := \frac{r_k(q-1) + r_k(q)}{R_k(q-1) + R_k(q)} - \frac{r_k(q)}{R_k(q)},\ q = 2, \ldots, N. \tag{14}$$

Thus, it is necessary to calculate the sequential disproportion of the SDF using integral disproportion. For example, for the red component of the k-th standard, if $p = 3$, in order to calculate the sequential integral disproportion, three steps should be performed:

Step 1. We calculate the integral disproportion of the red component $r_k(q)$ arriving at the recognition system with respect to the red component of the k-th standard image $R_k\ (q)$:

$$I_1(q) = @I^{(1)}_{R_k} r_k(q) = \frac{r_k(q-1) + r_k(q)}{R_k(q-1) + R_k(q)} - \frac{r_k(q)}{R_k(q)}$$

Step 2. The integral disproportion of $I_1(q)$ with respect to $R_k(q)$ is calculated:

$$I_2(q) = @I^{(1)}_{R_k} I_1(q) = \frac{I_1(q-1) + I_1(q)}{R_k(q-1) + R_k(q)} - \frac{I_1(q)}{R_k(q)}$$

Step 3. The integral disproportion of $I_2(q)$ with respect to $R_k(q)$ is calculated:

$$SDF(q) = I_3(q) = @I^{(1)}_{R_k} I_2(q) = \frac{I_2(q-1) + I_2(q)}{R_k(q-1) + R_k(q)} - \frac{I_2(q)}{R_k(q)} \tag{15}$$

If the disproportion (15) is equal to zero at the last step, this means that the corresponding pixel of the k-th standard image matches to the q-th pixel of the analyzed image.

However, in practice, the errors will accumulate when calculating the sequential disproportionality of SDF(q). As a result, at the last stage, a result close to zero, but isn't zero, can be obtained. Therefore, the result of the calculation at the last stage should be compared with some number ε close to zero. If modulo of this disproportion is less than or equal to ε, it is considered that the recognition has taken place.

6 The Recognition Algorithm

As an example, let's consider the recognition of fragments of standard images using the red component of the color of the pixels of the analyzed signal.

Algorithm

Step 1. Fix a (small) positive tolerance threshold. It will be used to compare the disproportion function's (absolute) value with this parameter for recognizing.
Step 2. Read the red brightness values of the pixels belonging to all standards.
Step 3. Select standard and read the red pixel components of analyze signal as $r_k(q)$, $1 \leqslant q \leqslant N$.
Step 4. Calculate the disproportion function SDF(q) of r_k with respect to R_k by (15) and store its values as the array

$$D_k^R(q) = SDF(q), q = 2, K, N$$

Step 5. The recognition test. If $\left|D_k^R(q)\right| \leq \varepsilon$ it means that pixel q of the analyzed image can be associated (identified) as the q-th pixel of standard (fragment) k. It is copied to pixel q of the result image.
Else if $\left|D_k^R(q)\right| > \varepsilon$, the appropriate pixel of result image remains empty. This procedure is repeated for each of pixel.
Step 6. Set $k = k + 1$. If $k \leq m$ go to Step 4, else go to Step 7.
Step 7. End and show the obtained image.

Remark 2. In practice, the tolerance parameter ε can be initially set equal to zero and then increased by and by in order to improve the recognition of the standards. That is, ε should be increased until fragments of the standard images begin to appear. In practice, the researcher can always assign the maximum value for ε depending on the problem being solved.

Obviously, as ε increases, the recognition becomes easier. However, it should be remembered that in doing so it becomes rougher. If the difference between the standard signals is within ε, then they become indistinguishable. The result may be false recognition.

7 Example of Recognition of a Distorted 2-D Image in the Absence of Interference

The system of technical vision is simulated, in which the color components of Red, Green and Blue are used without any preliminary processing.

We will also assume that the scales of the reference and analyzed images are the same and rotation is absent.

Suppose, that due to poor object lightning, most part of the signal falls into the brightness range from 1 to 15 (at a maximum of 255), which is called "Blacker Than Black" or BTB. Often it is not used, and, for example, for video amplifiers, synchronization pulses are located in the corresponding voltage range. However, the hardware settings allow, if necessary, to use this area [22].

The signal from the photodetector of the photodiode array is fed to the input of the transistor. Near the zero value of the input signal, the current – voltage characteristic (CVC) of the p – n junction has a nonlinear region [23].

We assume that the incoming signal does not go beyond the nonlinear portion of the current – voltage characteristic of the transistor.

With a decrease in the illumination level of the object, a proportional relationship is maintained between the brightness of the image entering the photodiode array and its standard [24]. However, after passing through non-linearity, the signal is distorted. The purpose of the simulation is to show the fundamental possibility of recognizing a fragment of a standard image from a distorted and attenuated signal.

The BTB part is about 6% of the entire brightness scale. We take the interval of the nonlinear portion of the current-voltage characteristic 5%.

In a normalized form, the signal at the input of the transistor can vary from 0 to 0.06 within the non-linear CVC, that is, from 0 to 0.05.

We consider the case when the nonlinear part of the current – voltage characteristic has the form Fig. 1.

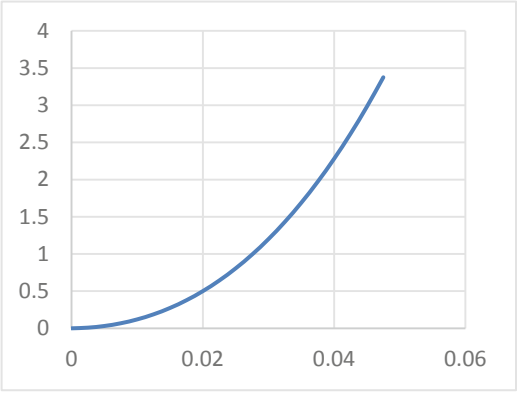


Fig. 1. The nonlinear part of the current-voltage characteristic of the p-n junction used in the example

It is described by a 3-rd order polynomial:

$$y = x + 1000x^2 + 10000x^3,$$

where x - is the relative value of the input signal (red component of the reference image);

y - is the parameter at the output of the nonlinear device.

This parameter is then reduced to the absolute values of the Red_new brightness so that the signal converted after the nonlinear device falls into the blacker region by multiplying, in this example, by 255/100. For them, the SDF- disproportion is calculated (15) in accordance with the above algorithm. When calculating it, pixels were excluded for which Red_new is zero or for which division by zero occurred. Allowable discrepancy $\varepsilon = 10^{-4}$.

Figure 2 shows the recognition results for $\varepsilon = 10^{-4}$ (top) and for $\varepsilon = 10^{-8}$ (bottom).

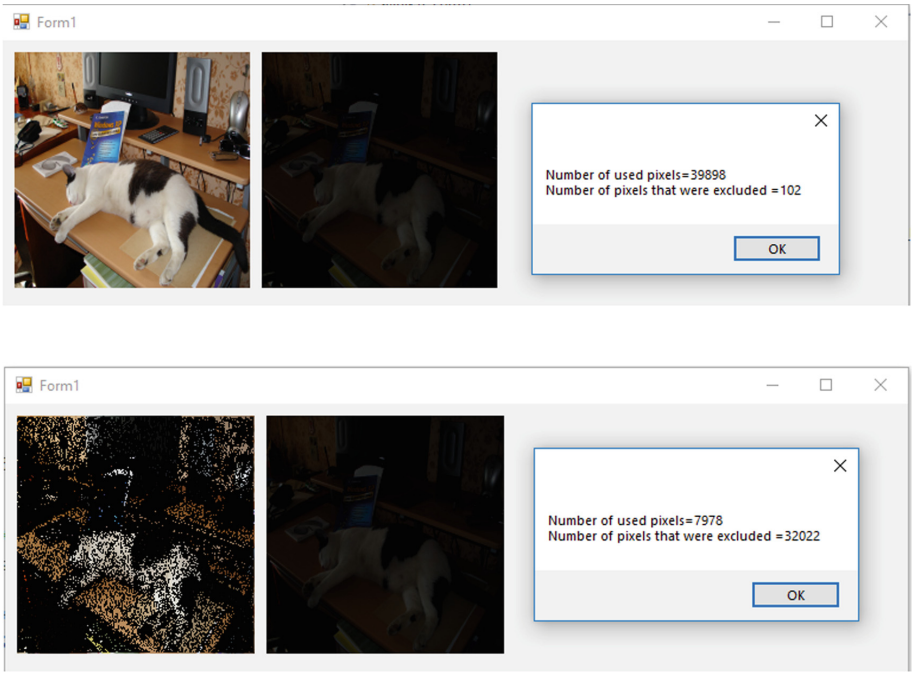


Fig. 2. Distorted and recognized images for $\varepsilon = 10^{-4}$ (top) and for $\varepsilon = 10^{-8}$ (bottom).

At the bottom of each picture, distorted signal that is received for recognition is shown. The recognized reference image is shown at the top of each picture. The number of pixels used and excluded from the analysis is also shown in each picture.

The comparison of the results shows that the decreasing of ε led to the worst darker recognized image. This shows the importance of choosing the right value of the tolerance threshold.

8 Recognition of Distorted Image with Presence of the Impulsive Interference

Consider the case when the signal η(t) is superimposed on the analyzed signal (5) and it is now described by the expression:

$$y(t) = a_p f_i^p(t) + a_{p-1} f_i^{p-1}(t) + \ldots + a_1 f_i(t) + \eta(t) \tag{16}$$

Naturally, now when calculating SDF (4), the component determined by the interference $\eta(t)$ will be added. In the common case, this component will not be equal to zero.

The algorithm proposed above makes it possible to automatically detect the fact that interference has disappeared and recognize the standard signals at this time. In the limit, even one pixel of standard can be recognized.

9 Conclusions

In this paper, we propose a method for the operational recognition of fragments of standard images, both continuous and discrete, based on the use of disproportion functions with respect to the first-order derivative. Recognized signals are distorted as a result of passing through nonlinear devices whose static characteristic is described by a polynomial with unknown coefficients. A recognition algorithm is proposed.

The proposed method is demanding on the speed and memory capacity of the computing device. However, it should be noted that the recognition process is easily parallelized. Indeed, each standard signal can be recognized in parallel. Moreover, for each of them, you can set a certain time shift between the analyzed and standard signals.

As a result, it becomes possible to recognize weak signals, the processing of which is impossible, for example, due to the nonlinear area of static characteristic of the amplifying device near the zero value of the input signal. In particular, using this method, it is possible to expand the working range of the brightness of the objects of vision systems by using a nonlinear portion of the current-voltage characteristic of the transistor in the region of zero. It is assumed that in these vision systems, R, G, B signals are processed without any correction, which is often used to bring the image color closer to the color perception of the human eye.

References

1. Barhatov, V.A.: Obnaruzhenie signalov i ih klassifikacija s pomow'ju raspoznavanija obrazov. Defectoscopiya **4**, 14–27 (2006)
2. Deglina, J.U.B.: Nejrosetevoj algoritm raspoznavanija signalov akusticheskoj jemissii. SHtuchnij intelekt **4**, 731–734 (2006)
3. Spravochnaja sistema po moduljam Digispot II. http//redmine.digispot.ru/…/digispot.ru/projects/digispot/wiki/WikiStart
4. Venediktov, M.D., Markov, V.V., Eydus, G.S.: Asinhronnyie adresnyie sistemyi svyazi. Svyaz, M. (1968)

5. Voishvillo, G.V.: Amplification devices Textbook for High School's.Radio and Communications, M. (1983)
6. Putjatin, E.P.: Normalizacija i raspoznavanie izobrazhenij. http://sumschool.sumdu.edu.ua/is-02/rus/lectures/pytyatin/pytyatin.htm
7. Bezruk, V.M., Ivanenko, S.A.: Selection and recognition of the specified radio signals in the SW band. J. Inf. Telecommun. Sci. **9**(2), 21–25 (2018)
8. Shevchuk, B.M., Zadiraka, V.K., Fraier, S.V., Luts, V.K.: Operatyvne rozpiznavannia frahmentiv i kompleksiv syhnaliv ta vydilennia obiektiv videodanykh zasobamy obiektnykh system bezprovidnykh merezh. J. Iskusstvennyiy intellekt **3**, 275–283 (2013)
9. Muryigin, K.V.: Obnaruzhenie avtomobilnyih nomernyih znakov s ispolzovaniem predvaritelnoy obrabotki kandidatov. J. Iskusstvennyiy intellekt **3**, 193–199 (2013)
10. Artificial Neural Networks: Concepts and Theory. IEEE Computer Society Press (1992)
11. Osovskij, S.: Nejronnye seti dlja obrabotki informacii. Finansy i statistika, M. (2004)
12. Ayinde, B.O., Inanc, T., Zurada, J.M.: Regularizing deep neural networks by enhancing diversity in feature extraction. IEEE Trans. Neural Netw. Learn. Syst. **30**, 2650–2661 (2019)
13. Lazorenko, O.V., Lazorenko, S.V., CHernogor, L.F.: Primenenie vejvlet-analiza k zadache obnaruzhenija sverhshirokopolosnyh signalov na fone pomeh. Radiofizika i radioastronomija **1**(7), 46–63 (2002)
14. Avramenko, V.V., Slepushko, N.JU.: Raspoznavanie jetalonnyh signalov pri nepolnoj informacii o harakteristikah pomeh. Visnik SumDu **4**, 13–18 (2009)
15. Avramenko, V.V.: Harakteristiki neproporcional'nostey i ih primeneniya pri reshenii zadach diagnostiki. Vestnik SumGU **16**, 12–20 (2000)
16. Kalashnikov, V., Avramenko, V.V., Demianenko, V.N., Kalashnykova, N.: Fragment-aided recognition of images under poor lighting and additive impulse noises. Procedia Comput. Sci. **162**, 487–495 (2019)
17. Karpenko, A.P.: Integral'nye harakteristiki neproporcional'nosti chislovyh funkcij i ih primenenie v diagnostike. Vestnik SumGU **16**, 20–25 (2000)
18. Kalashnikov, V.V., Avramenko, V.V., Kalashnykova, N.I.: Derivative disproportion functions for pattern recognition. In: Watada, J., Tan, S.C., Vasant, P., Padmanabhan, E., Jain, L.C. (eds.) Unconventional Modelling, Simulation, and Optimization of Geoscience and Petroleum Engineering, pp. 95–104. Springer, Heidelberg (2018)
19. Kalashnikov, V.V., Avramenko, V.V., Slipushko, N.Y., Kalashnykova, N.I., Konoplyanchenko, A.E.: Identification of quasi-stationary dynamic objects with the use of derivative disproportion functions. Procedia Comput. Sci. **108(C)**, 2100–2109 (2017)
20. Kalashnikov, V.V., Avramenko, V.V., Kalashnykova, N.I., Kalashnikov Jr., V.V.: A cryptosystem based upon sums of key functions. Int. J. Comb. Optim. Probl. Inform. **8**, 31–38 (2017)
21. Kalashnykova, N., Avramenko, V.V., Kalashnikov, V.: Sums of key functions generating cryptosystems. In: Rodrigues, J.M.F., Cardoso, P.J.S., Monteiro, J., Lam, R., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J.J., Sloot, P.M.A. (eds.) ICCS 2019. LNCS, vol. 11540, pp. 293–302. Springer, Cham (2019)
22. Jack, K.: Video Demystified. A Handbook for the Digital Engineer (5th Edition) (2007)
23. Tereschuk, R.M., Tereschuk, K.M., Sedov, S.A.: Poluprovodnikovyie priemno_usilitelnyie ustroystva, spravochnik radiolyubitelya, Kiev, «Naukova dumka» (1989)
24. Luzin, V.I., Nikitin, N.P.: Osnovyi televideniya. Uchebnoe elektronnoe tekstovoe, izdanie GOU VPO UGTU-UPI (2008)

Vehicle Detection and Classification in Difficult Environmental Conditions Using Deep Learning

Alessio Darmanin(✉), Hossein Malekmohamadi, and Abbes Amira

Institute of Artificial Intelligence, De Montfort University, Leicester, UK
alessio.darmanin@gmail.com,
{hossein.malekmohamadi,abbes.amira}@dmu.ac.uk

Abstract. The time drivers spend stuck in traffic is increasing annually, on a global level. Time lost in traffic imposes costs both economically and socially. Tracking congestion throughout the road network is critical in an Intelligent Transportation System (ITS), of which vehicle detection is a core component. Great strides have been made in deep learning over the last few years particularly with the convolutional neural network (CNN), a deep learning architecture for image recognition and classification. One area in image recognition where the use of CNN has been studied is vehicle detection. This paper explores an area of vehicle detection where a little study has been made, that is the detection and classification of vehicles in difficult environmental conditions. The purpose of this paper is to build a CNN able to detect vehicles from low resolution, highly blurred images in low illumination and inclement weather and classify the vehicles in one of five classes. The final model built in this paper is able to achieve 92% classification accuracy on images in difficult environmental conditions. This model can be deployed to a smart traffic management system.

Keywords: Deep learning · Convolutional neural network · Vehicle detection · Vehicle classification · Traffic video analysis

1 Introduction

Global traffic is an increasing problem across the globe. INRIX, Inc., the world leader in mobility analytics and connected car services [1], in a 2018 study of global vehicular traffic revealed staggering figures. In the United States, drivers in Boston and Washington D.C. spent up to 164 and 155 hours respectively sitting in traffic in 2018. Nationwide, drivers lost 97 hours in congestion costing US$87 billion in lost productivity, with an average of US$1,348 per driver [2]. The picture is the same for Europe [2]. In the United Kingdom, drivers on average spent 178 hours stuck in traffic, costing the country US$10.3 billion, or US$1,725 per driver. Drivers in Germany lost an average of 120 h due to congestion in 2018, costing the country US$5.8 billion, or US$1,203 per driver. Traffic will continue to increase. The U.S. federal highway administration (FHWA) projects the total Vehicle Miles Traveled (VMT) by all vehicle types to grow at an average rate of 1.1% annually over the 20 years through 2037, a total increase of 23% on current figures [3].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 686–696, 2021.
https://doi.org/10.1007/978-3-030-55180-3_52

Malta, the country on which this investigation is based, is one of the most traffic-congested places in Europe. Despite having a population of fewer than 500,000 people, Malta currently has the second-highest number of cars per capita in the European Union – nearly two passenger cars for every three inhabitants. The average commuter spends around 52 h a year in gridlock, despite the average commute in Malta being less than six kilometres [4], and traffic congestion is expected to cost Malta €317 million [5] in 2020, a relatively large proportion of Malta's GDP which is around €10 billion.

This paper uses images captured from a publicly accessible online traffic monitoring webcam. The images are used to train a convolutional neural network (CNN) to detect and classify vehicles into one of five classes. Images are low resolution, and those captured in the low illumination of early morning or late afternoon hours are highly blurred, as are those captured in inclement weather. These images represent conditions which a smart traffic management system, also referred to as Intelligent Transportation System (ITS), should be capable of handling. An ITS would record data such as vehicle count, travel speed, vehicle class and could help to minimise traffic problems by predicting potential congestion and redirect traffic through the use of electronic road signs.

The paper is organized as follows. Section 2 describes different technologies making up the current state of the art in vehicle detection. Section 3 analyses traffic and environmental conditions, and the vehicle images captured in these conditions. Section 4 discusses the convolutional neural network architecture and its implementation. Section 5 presents training results and testing. Section 6 discusses limitations and future work, and Sect. 7 puts forward final conclusions.

2 Current State of the Art

Vehicle detection and traffic surveillance technologies are core components of an ITS. Methods used for vehicle detection and traffic surveillance are based on sensor technologies, machine learning techniques, or a combination of sensor technologies and machine learning techniques. Sensor technologies are classified as either intrusive, non-intrusive or off-roadway [6].

In [7] the authors propose the development of a portable roadside magnetic sensor system for vehicle counting, classification, and speed measurement. The research carried out in [8] uses a dynamic traffic simulator to generate flows using historical traffic data from road sectors in the network equipped with sensors. The study made by the authors of [6] report on the development and implementation of an innovative smart wireless sensor for traffic monitoring, called intelligent vehicle counting and classification sensor (iVCCS). In [9] a video analysis method for counting vehicles is proposed, using adaptive bounding boxes to detect and track vehicles according to their estimated distance from the camera given the scene-camera geometry. The system described in [10] is a vision-based system and uses dedicated cameras already installed on road networks.

The technique adopted in [11] uses a large diverse dataset of cars created from overhead images. This dataset is used for classification and detection using a neural network called ResCeption. A deep learning approach used in [12] proposes a system to detect and count the number of vehicles in traffic surveillance videos based on the use of a Fast Region-based Convolutional Network (Fast R-CNN). The technique proposed in

[13] uses a method to detect vehicles from images based on road direction; it utilizes 3D car models to generate car poses and groups them into four pairs of viewpoint orientation. A different and innovative machine learning technique shown in [14] is based on mobile data usage to measure traffic flow and the prediction of traffic flow changes over time, using a combination of data mining and machine learning techniques.

The approach taken in [15] is to collect data from traffic sensors and process the data using machine learning, where traffic volume, speed, and road occupancy are used to predict short-term traffic flow. A different approach taken in [16] is to use data collected from multiple sensor sources before processing using deep learning. The authors propose a deep-learning-based traffic flow prediction method using a stacked autoencoder model. The researchers in [17] propose a hybrid modelling method that combines an ANN ensemble and a simple statistical approach to provide one-hour forecasts of urban traffic flow rates. Another deep learning technique can be seen in [18], where a deep neural network is used based on long short-term memory (LSTM) units. The deep LSTM model is used to forecast peak-hour traffic and to identify unique characteristics of the traffic data. The approach taken in [19] is to use a learning-based aesthetic model to estimate the state of traffic on traffic surveillance video. The training dataset consists of images obtained from traffic surveillance cameras.

Intrusive road sensors have the disadvantages of high installation costs and traffic disruption when carrying out installation, maintenance and repairs; non-intrusive sensors provide intrusive sensors' functionality with fewer difficulties, but they in turn are highly affected by climate conditions [20]. The machine learning techniques reviewed in this paper detect vehicles and estimate traffic flow in normal traffic scenarios only; those not affected by environmental conditions such as in [14] detect but do not classify vehicles.

3 Background

During the autumn and winter seasons in Malta, early morning and late afternoon rush hours occur when ambient light is low and with the additional possibility of rain. Figure 1 is an image showing heavy traffic during afternoon rush hour and the light conditions at that time of day. Images are captured from a publicly accessible traffic camera [21] on a busy main road in Malta which experiences heavy traffic during rush hour.

Figure 2 shows changing light conditions during morning rush hour; vehicles appear blurred in images from 06:00 up to 07:45. Figure 3 shows changing light conditions during afternoon rush hour; vehicles appear blurred in images from 16:30 onwards.

4 Proposed Method

Great strides have been made in deep learning over the last few years, leading to very good performance in diverse areas such as visual recognition, speech recognition and natural language processing, with convolutional neural networks being one of the deep learning architectures most extensively studied [22].

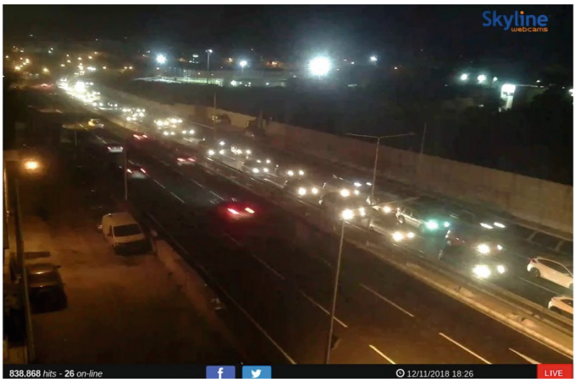


Fig. 1. Image showing traffic and light conditions during afternoon rush hour [21]

Fig. 2. Images showing changing light conditions during morning rush hour; these images were captured between 06:00 (top left corner image) and 08:00 (bottom right corner image).

In this paper a Convolutional Neural Network (CNN) is built and trained to successfully:

- detect and classify vehicles from low-resolution images
- detect and classify vehicles from images where the vehicle is blurred
- detect and classify vehicles from images in varying environmental conditions, such as very low illumination due to time of day or rainy weather
- detect and classify vehicles into one of five classes: *Bus, Car, Motorbike, Truck, Van*.

The training dataset consisted of *Background*, *Bus*, *Car*, *Motorbike*, *Truck* and *Van* images:

- 107,000 images used for training
- 17,000 images used for validation.

Fig. 3. Images showing changing light conditions during afternoon rush hour; these images were captured between 16:30 (top left corner image) and 18:30 (bottom right corner image).

The *Background* class is used to denote a lack of vehicle present in the image.

The training dataset included vehicle images from the time of day where illumination is very low, that is between 06:00 and 07:45 and between 16:30 and 18:30, as well as images captured during rainy weather.

4.1 Architecture

Figure 4 depicts the CNN architecture used in this study. The model has over 34.5 million parameters. The CNN is based on a modified VGG-16 model [23], with the following changes:

- added batch normalization layers
- added L2 regularization
- used different filter sizes in the first four convolutional layers
- changed to 'no padding' from the third to the thirteenth convolutional layer.

The CNN included five batch normalization layers [24]; dropout layers were only applied after all batch normalization layers, as recommended in [25] where improvements in prediction accuracy were observed when using this approach.

To reduce overfitting, regularization techniques were used. These included random data augmentation so that the CNN would see different versions of an image in subsequent epochs using label-preserving transformations [26], Dropout [27] applied to the fully-connected layers, L2 weight decay parameter regularization applied to the first convolutional layer, and early stopping.

4.2 Data Input, Normalization and Real-Time Random Augmentation

The input to the CNN was a fixed 240 $\times$ 240 pixel size RGB image, with each image being a portion of the image shown in Fig. 1 captured from a traffic camera [21].

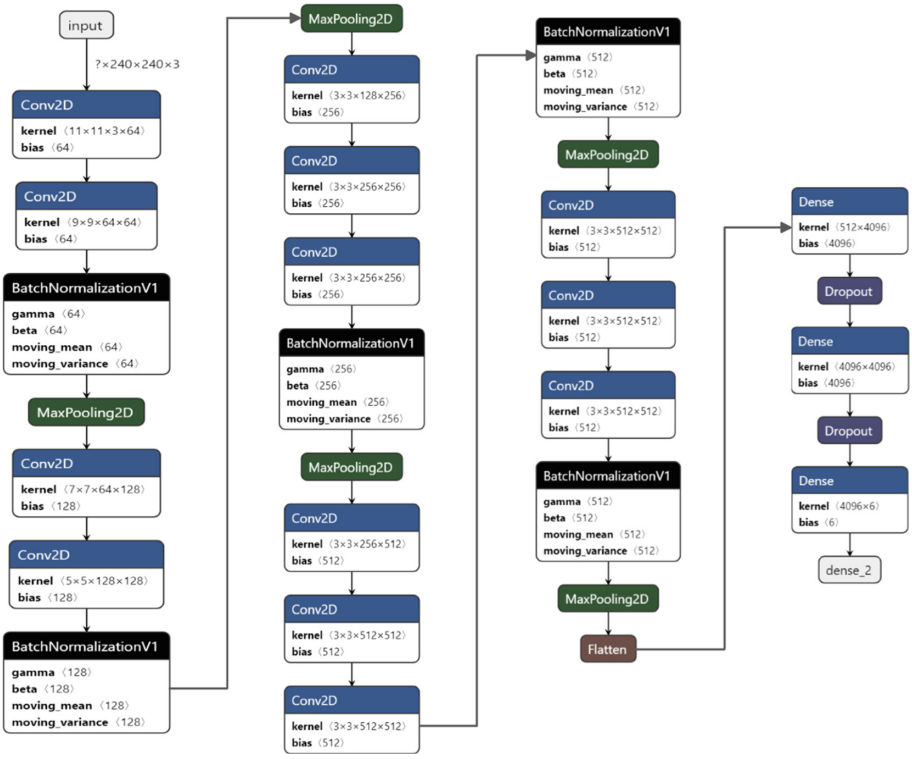


Fig. 4. Convolutional Neural Network.

One of the following forms of augmentation was randomly performed on the image before input to the CNN:

- horizontal flip
- rotation: rotate the image by up to 20°
- zoom: zoom out or zoom in by a factor of between 0.8x to 1.2x, respectively.

5 Training Results and Testing

Training different CNN models was an iterative approach till the hyperparameters which gave the best performance, in terms of training, validation and test accuracy, were identified. The number of epochs set for training was 250 with mini batch size of 32, initial learning rate was 0.001, L2 Regularization was 0.0005 and dropout was 0.5.

The CNN was then finetuned twice to achieve better performance. In the first finetuning iteration the first 7 convolutional layers were frozen and the CNN retrained using a learning rate of 0.0001. In the second finetuning iteration the first 10 convolutional layers were frozen and the CNN retrained using a learning rate of 1×10^{-6}. The best performing CNN model achieved training accuracy of 97.3% and validation accuracy of 97.0%.

The CNN was tested on 350 vehicle images covering the morning rush hour. These consisted of 140 images in a normal traffic scenario and 210 images in difficult environmental conditions. The latter included images from the time of day where illumination is very low, between 05:45 and 07:45, and images in rainy weather. Table 1 shows the classification metrics on this test data; average precision was 93%.

Table 1. Classification report showing classification metrics on test data.

Class	Precision	Recall	F1-score	Support
Background	0.87	1.00	0.93	27
Bus	0.96	0.96	0.96	26
Car	0.92	0.96	0.94	98
Motorbike	0.92	0.85	0.88	27
Truck	0.93	0.94	0.94	90
Van	0.96	0.88	0.92	82

The lowest performing class was the *Motorbike* class with an F1-score of 0.88; the other classes achieved an F1-score of 0.92 or better.

Table 2 shows the accuracy by class for all images of vehicles in the test data with difficult environmental conditions. From the 210 test images with difficult environmental conditions 193 were predicted correctly, resulting in an accuracy of 92%. Lowest accuracy was in the Motorbike class with 64%. This can be explained by the fact that it is the class with the lowest number of images in the training set. Better accuracy can be achieved through additional motorbike images, depicting difficult environmental conditions, in the training dataset.

Table 2. Accuracy by class on difficult images in the test data.

Class	Correct	Actual	Accuracy
Background	17	17	100%
Bus	18	18	100%
Car	56	59	95%
Motorbike	7	11	64%
Truck	50	53	94%
Van	45	52	87%
Total:	**193**	**210**	**92%**

Figure 5 shows a car which was incorrectly classified as a motorbike. The training dataset did not include similar cars, however there were quadbikes which were labelled as motorbikes. The misclassified car is similar in shape and size to a quadbike.

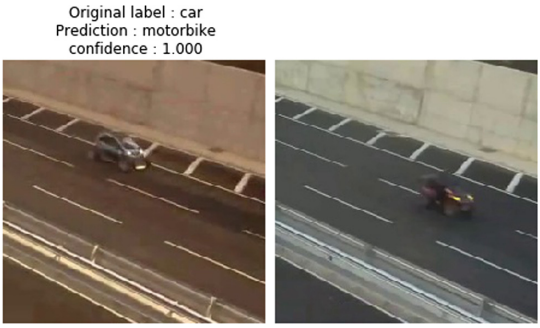


Fig. 5. Left image is a car incorrectly classified as motorbike. Right image is a quadbike from the training dataset.

Figure 6 shows two samples of the Motorbike class which were incorrectly classified as Background. The motorcyclist cannot be seen in any of the images, as illumination is too low. The streak made by the motorbikes' headlamps is right on top of the white road markings. This could have been the cause of the misclassification.

Fig. 6. Two motorbikes which were not classified correctly.

Figure 7 shows some images from the test dataset which were classified correctly. They are images from the test data with difficult environment conditions, for the time period between 05:45 and 07:45, and also include two images captured in rainy weather.

6 Limitations and Future Work

Current limitations of the CNN proposed in this paper include: training data, which is extracted from only one traffic camera source and thus shows vehicles being driven in only one direction; detection and classification, which is only carried out on images displaying a single vehicle. Future work will focus on additional training data with images from other traffic cameras; extending the model to use an object detection approach similar to [28–30] allowing detection, classification and counting of multiple vehicles present in an image; deploying the model to a smart traffic management system capable of classifying traffic density.

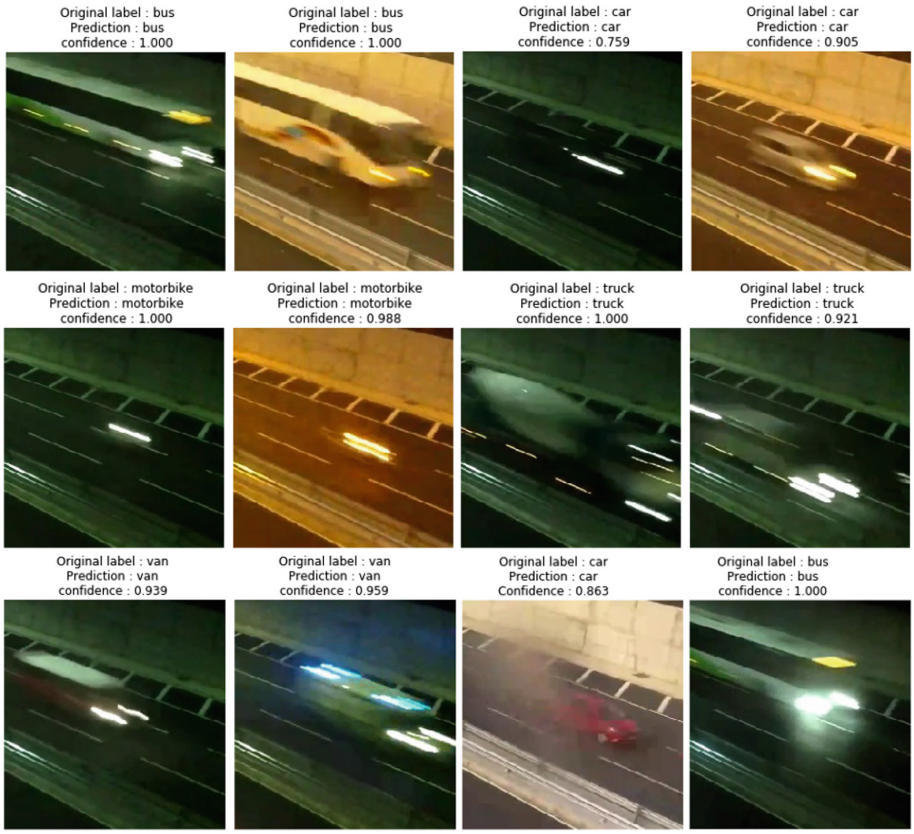


Fig. 7. Images in difficult environmental conditions correctly classified by the CNN. The first 10 images were captured in low illumination. The last 2 images were captured in rainy weather.

7 Conclusion

In this paper, a CNN is implemented which achieved 93% average precision on test data. The CNN classified images in difficult environmental conditions with 92% accuracy. To the author's knowledge, no studies have been carried out for the detection and classification of vehicles in environmental conditions like those covered in this paper. The solution presented in this paper could form part of an ITS helping in traffic flow analysis and traffic management planning in cities by being able to detect and classify vehicles in all environmental conditions including very low light conditions like those shown in Fig. 1. Doing this would help in traffic management by predicting potential congestion and redirecting traffic through the use of electronic road signs. Vehicle classification would also benefit pollution management by setting a limit, for example, on the number of trucks passing through particular sections of the road network. With traffic flow data being captured using low-cost internet cameras, and with high bandwidth internet increasingly widespread, the relative ease and low cost of implementing the hardware

would allow this system to be implemented in larger numbers and thus allow potential city-wide coverage.

References

1. International Parking & Mobility Institute. https://www.parkingmobility.org//2019/02/13/member-news-inrix-ranks-most-congested-cities-in-theworld-inrix-2018-global-traffic-scorecard-just-released/. Accessed 31 Jan 2020
2. INRIX Global Traffic Scorecard (2018). https://static.poder360.com.br/2019/02/INRIX_2018_Global_Traffic_Scorecard_Report__final_.pdf. Accessed 31 Jan 2020
3. U.S. Department of Transportation, https://www.fhwa.dot.gov/policyinformation/tables/vmt/vmt_forecast_sum.pdf. Accessed 31 Jan 2020
4. Equal Times. https://www.equaltimes.org/can-malta-tackle-its-traffic. Accessed 31 Jan 2020
5. Institute For Climate Change And Sustainable Development. https://ec.europa.eu/malta/sites/malta/files/docs/body/study_on_traffic_online.pdf. Accessed 31 Jan 2020
6. Balid, W., Tafish, H., Refai, H.H.: Intelligent vehicle counting and classification sensor for real-time traffic surveillance. IEEE Trans. Intell. Transp. Syst. **19**(6), 1784–1794 (2018)
7. Taghvaeeyan, S., Rajamani, R.: Portable roadside sensors for vehicle counting, classification, and speed measurement. IEEE Trans. Intell. Transp. Syst. **15**(1), 73–83 (2014)
8. Abadi, A., Rajabioun, T., Ioannou, P.A.: Traffic flow prediction for road transportation networks with limited traffic data. IEEE Trans. Intell. Transp. Syst. **16**(2), 653–662 (2015)
9. Baş, E., Tekalp, A.M., Salman, F.S.: Automatic vehicle counting from video for traffic flow analysis. In: IEEE Intelligent Vehicles Symposium 2007, pp. 392–397 (2007)
10. Crouzil, A., Khoudour, L., Valiere, P., Truong Cong, D.N.: Automatic vehicle counting system for traffic monitoring. J. Electron. Imaging **25**(5), 1–12 (2016)
11. Mundhenk, T.N., Konjevod, G., Sakla, W.A., Boakye, K.: A large contextual dataset for classification, detection and counting of cars with deep learning. In: European Conference on Computer Vision 2016, pp. 236–251 (2016)
12. Zhang, Z., Liu, K., Gao, F., Li, X., Wang, G.: Vision-based vehicle detecting and counting for traffic flow analysis. In: Proceedings of International Joint Conference on Neural Networks, October 2016, pp. 2267–2273 (2016)
13. Prahara, A., Murinto: Car detection based on road direction on traffic surveillance image. In: Proceeding of 2nd International Conference on Science in Information Technology. ICSITech 2016 Inf. Sci. Green Soc. Environ., pp. 344–349 (2017)
14. Saliba, M., Abela, C., Layfield, C.: Vehicular traffic flow intensity detection and prediction through mobile data usage. In: CEUR Workshop Proceedings 2018, pp. 66–77 (2018)
15. Mohammed, O., Kianfar, J.: A machine learning approach to short-term traffic flow prediction : a case study of interstate 64 in Missouri. In: 2018 IEEE International Smart Cities Conference (ISC2), pp. 1–7 (2018)
16. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2014)
17. Moretti, F., Pizzuti, S., Panzieri, S., Annunziato, M.: Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling. Neurocomputing **167**, 3–7 (2015)
18. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: a generic approach for extreme condition traffic forecasting. In: Proceeding sof 17th SIAM International Conference on Data Mining SDM 2017, pp. 777–785 (2017)
19. Shi, X., Shan, Z., Zhao, N.: Learning for an aesthetic model for estimating the traffic state in the traffic video. Neurocomputing **181**, 29–37 (2016)

20. Guerrero-Ibáñez, J., Zeadally, S., Contreras-Castillo, J.: Sensor technologies for intelligent transportation systems. Sensors (Switzerland) **18**(4), 1–24 (2018)
21. Skyline webcams. https://www.skylinewebcams.com/en/webcam/malta/malta/traffic/traffic-cam2.htm. Accessed 15 Sept 2019
22. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Chen, T.: Recent Advances in Convolutional Neural Networks. arXiv:1512.07108v6 [cs.CV] (2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV] (2015)
24. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: 32nd International Conference on Machine Learning ICML 2015, pp. 448–456. International Machine Learning Society (IMLS) (2015)
25. Li, X., Chen, S., Hu, X., Yang, J.: Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift. arXiv:1801.05134v1 [cs.LG], (2018)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. NIPS (2012)
27. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580v1 [cs.NE] (2012)
28. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv:1804.02767v1 [cs.CV] (2018)
29. Liu, W., et al.: SSD: single shot MultiBox detector. In: European Conference on Computer Vision 2016, pp. 21–37. Springer (2016)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Analy. Mach. Intell. **39**(6), 1137–1149 (2017)

Tree-Structured Channel-Fuse Network for Scene Parsing

Ye Lu[1], Xian Zhong[1,2], Wenxuan Liu[1], Jingling Yuan[1,2(✉)], and Bo Ma[3]

[1] School of Computer Science and Technology,
Wuhan University of Technology, Wuhan, China
224625@whut.edu.cn
[2] Hubei Key Lab of Transportation Internet of Things,
Wuhan University of Technology, Wuhan, China
[3] Department of Computer and Information Science and Engineering,
University of Florida, Gainesville, USA

Abstract. Scene parsing requires effectively discovering context information and the structure of the network plays a critical role in the task. To handle the problem of segmentation objects at multiple scales, we proposed a tree-structured channel-fuse network (TCFNet) to obtain more representative information. In detail, we create a tree structure to merge the multiple-level feature maps, which are fused by the channel-fuse module, with multi-scale context information in a hierarchical way. And the module refines the feature maps during the process of propagating context between channels. The proposed TCFNet achieves impressive results on Cityscapes validation set and test set, verify the effectiveness of our proposed approach.

Keywords: Scene parsing · Tree-structured · Channel fuse

1 Introduction

Scene parsing is a significant and challenging task in semantic segmentation, whose goal is to apply an appropriate label for each pixel in scene images. It takes advantage of semantic information to split the image into blocks. It helps us understand the scene better and plays a key part in many fields such as robot sensing, automatic driving and so on.

One main difficulty in scene parsing is the existence of objects at multiple scales. There are mainly two structures to overcome it. As showed in Fig. 1(a), one way [3,34] is to add multiple branch with different scales context information after the backbone network. PSPNet [34] create multiple branches by adopting pyramid pooling module. However, it only takes advantage of final output of the backbone network. The low-level feature maps with small receptive fields and high resolution, which restore more details and contain more small scale context, are not used in PSPNet. While another structure, the U-shape structure [20,28], overcomes this problem. The skip connection is widely adopted to make the shortcut between high-level and low-level feature maps in the U-shape structure,

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 697–709, 2021.
https://doi.org/10.1007/978-3-030-55180-3_53

illustrated in Fig. 1(b). While in UNet [20], the higher feature map utilizes the lower feature map and the lower feature map is coarser though contains more low context information. UNet performs well in the medical dataset since the medical images are relatively simple in semantic, which are more sensitive to the low-level information such as edges and lines. In contrast, for the scene parsing dataset, the semantic information is richer and more abstract. The unprocessed low-level information not only has little effect, but also introduces a lot of noise.

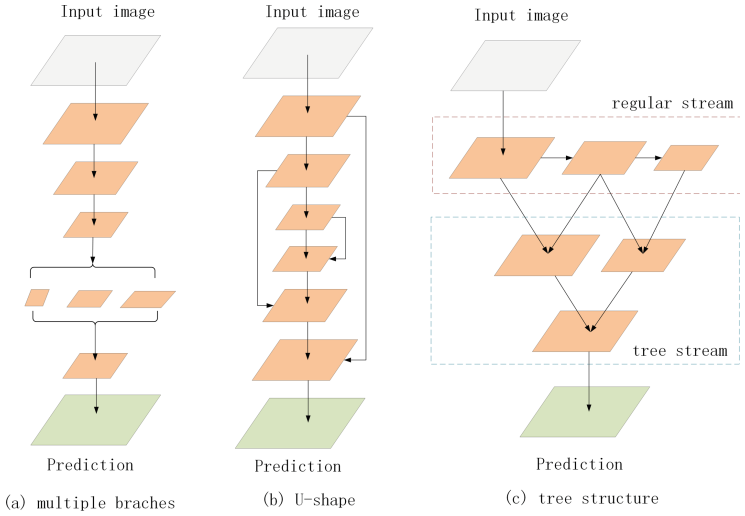


Fig. 1. Illustration of the structure to handle multiple scales. (a) presents the way to add multiple branches with different scale context information. (b) indicate the U-shape structure which widely uses the skip connection to associate low-level and high-level features. (c) is our proposed tree structure. The regular stream is framed by a red border whose goal is to generate multi-level features. The tree stream contains many repeated sub-tree to fuse the adjacent features in a hierarchical way.

To handle the problem of segmentation objects at multiple scales, we proposed a tree-structured channel-fuse network (TCFNet). TCFNet creates the channel-fuse module to fuse multiple input feature maps, different from above methods that take simple add or concatenate operation. This module split the input feature maps into multiple groups and each group generates the output considering adjacent groups, which harvests more precise context information. To overcome the shortage of U-shape mentioned above, we take adjacent feature maps output by the inner layers of the backbone network as input and use the channel-fuse module to fuse them together in a hierarchical way, which constitutes a tree structure. So as showed in Fig. 1(c), we use both low-level and high-level feature maps with multiple context information as input and reduce the number of feature maps layer by layer in tree stream, which harvest more representative information to obtain a coarse-to-fine result.

We have conducted extensive experiments to show the performance of our TCFNet. Our proposed TCFNet with ResNet-101 as backbone achieves superior performance on the challenging scene parsing dataset, Cityscapes. In summary, our main contributions are threefold:

• We introduce a novel tree-like structure in this work, which effectively captures the multi-scale context information from the different stage features in a hierarchical way.

• We create a channel-fuse module by propagating the information along the channels to harvest more representative features.

• We achieve superior results on the validation set and test set in Cityscapes, obtaining the result of 79.7% on the Cityscapes test dataset.

2 Related Work

2.1 Scene Parsing

Recently, many methods [2–4] based on CNN have sprung up and achieve great success in scene parsing. FCN [18] is the first network which extends image-level classification to pixel-level classification. Following this work, many FCN-based methods have achieved remarkable progress in scene parsing. RefineNet [15], DFN [29] adopt the encoder-decoder structure to do the dense prediction task. Unet [20] concatenates the output of the front layer and the back layer together to adaptively choose the representations. SAC [32] and Deformable Convolutional Networks [7] improve the traditional convolution operation to handle the challenge in scene parsing like deformation. Most methods focus on two main challenges in scene parsing, (i) multi-scale: the scene images contain various object with many different scales, which is obvious in the scene parsing dataset Cityscapes. (ii) context: scene parsing is aiming to predict every pixel in the image, so it is of vital importance to fully utilize the connection between the images to do the dense prediction. To handle the above problems, Deeplabv3 [3], Deeplabv3+ [4] all use atrous convolution to gain bigger receptive fields. Pyramid pooling is widely used in many networks like PSPNet [34], which is proved to be useful to handle the multi-scale problem. GCN [11] adopts global convolution module to gain richer context information. Recently, HRNet [23] restores rich context information in the long range of the network by taking advantage of high resolution. Inspired by previous work, we adopt dilated convolution in our TCFNet. In addition, we keep the resolution of the high resolution in our Tree Stream and obtain the 1/4 of the input image's size. Our model not only takes advantage of multi-scale input feature maps, but also gains richer context information.

2.2 Context

Discovering richer contextual information is the most vital goal for scene parsing. These methods concentrate on three aspects including spatial context, channel context and stage context. The spatial context is commonly used to boost

the model performance, including global spatial context and multi-scale spatial context. ParseNet [17] fuse all pixels and thus gain a wider receptive field to have much more spatial context. Ma et al. [19] propose a similarity method to effectively harvest context information. Chen et al. [2] utilize atrous convolution and collect long range of information in the feature map. OCNet [31] introduce the attention mechanism to calculate the similarities between all the pixels. Other methods focus on multi-scale spatial context. PSPNet [34] pools multi-scale region to generate different scales of context information. Similarly, ASPP in DeepLabV3 [3] employs atrous convolution with different rates, which enables the network gaining wider and multi-scale features. The channel context is also widely used in scene parsing. ShuffleNet [33] apply channel shuffle operation to consider channel contextual information relevance. SENet [8] add another squeeze and excitation layer and assign different weights to different channels. BiseNet [28] further uses channel context both in ARM module and FFM module. Recently, more and more methods prefer to fuse low layer and high layer information. Low layer tends to have some information like shape, size, color and so on. While some high layer includes much more context information learned by deep convolutional neural network (DCNN). Deeplabv3 uses context information generated by stage2 in ResNet. HRNet [23] fuse the different scale resolution feature map, gaining good performance in scene parsing.

2.3 DCNN Structure

The structure is critical for DCNN to achieve better performance in various computer vision tasks. The early networks mostly have the linear structure like LeNet [13], AlexNet[22]. With the development of calculation ability, the networks became deeper and the networks with deeper structure tend to perform better in various tasks. VGG [5] has better performance compared to AlexNet [12]. The GoogLeNet [24] and Incepton [25] add additional paths to the network structures. Networks' structures have some no-linearly parts in them. To handle the gradient vanishing problem, residual block constructs some skip the path to make the network deeper. ResNeXt [27] combines the previous work and it broadens the width of the ResNet. The networks become deeper and wider to improve performance on a variety of visual problems [36]. Recently, Unet [20] adopts an u-shape structure to fuse low-level and high-level features. DenseNet [9] widely use the skip structure and fuse the features from the previous stages. HRNet [23] creates parallel branches with different resolutions. The networks become deeper, wider and more non-linear. Inspired by the development of these networks, we propose a tree-like structure to fuse the feature maps made by two adjacent stages in ResNet.

3 Approach

In this section, we will introduce our network in detail. Firstly, we present our TCFNet for scene parsing, as illustrated in Fig. 2(c). Then we formulate the information propagation between channels in our channel-fuse module.

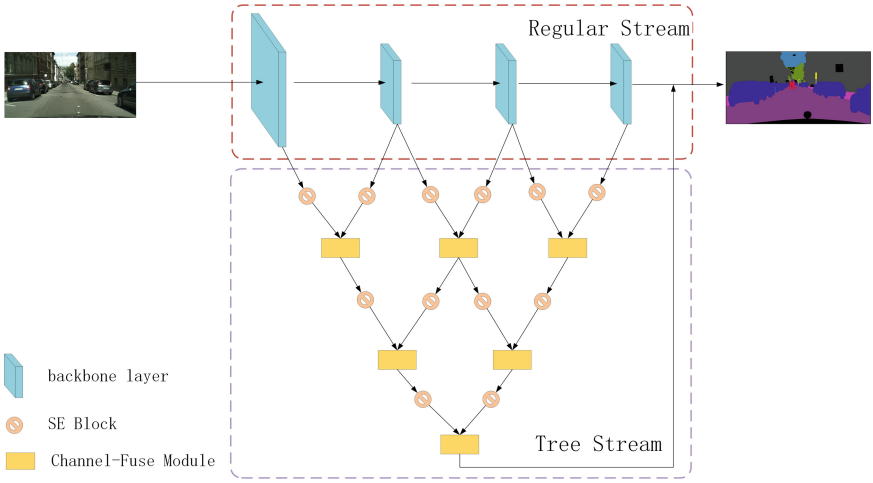


Fig. 2. Architecture of TCFNet contains two main streams. The regular stream and the tree stream. The regular stream is usually the backbone network like ResNet-101. The tree stream is a tree-like structure focusing on fusing different stage feature maps generated by regular stream through SE blocks and channel-fuse modules. The SE block in tree stream aims to refine the input feature maps and the channel-fuse module fuse two refined feature maps by exploiting the context between channels.

3.1 Overall

The framework of our proposed TCFNet is shown in Fig. 2. Our network is composed of two streams. The first stream is called regular stream and it is a standard segmentation CNN like ResNet-101. It is used for generating multi-scale feature maps with different resolutions or channels. Then the generated feature maps are delivered to the tree stream. The tree stream consists of some basic repeating sub-trees to fuse neighboring feature maps in upper layers of our TCFNet.

Regular Stream. Regular stream, denoted as $R(I)$, takes image $I \in \mathbb{R}^{H \times W \times 3}$ with height H and width W as input. It produces dense pixel features X_i with the spatial size of $H_i \times W_i$ after stage i in ResNet. In order to retain more context information, we remove the last two down-sampling layers and replace them with dilated convolutions, just like the modifications in PSPNet. So the size of output X_4 of the last stage is $1/8$ of the input image I. Generated output feature maps X_i are as given as following,

$$X_i = \begin{cases} \phi(I) & \mathrm{i} = 1 \\ \phi(X_{i-1}) & \mathrm{i} = 2, ..., \mathrm{N} \end{cases} \tag{1}$$

where ϕ denotes several convolution operations in the backbone network. Output X_i is also the input of tree stream.

Tree Stream. Tree stream, denoted as $T(X)$, fuses the neighboring features T_{i-1}^{j-1} and T_i^{j-1} in upper level $j-1$ by a repeating basic sub-tree and generates a feature map T_i^j, which is formulated as:

$$T_i^j = F(S(T_{i-1}^{j-1}), S(T_i^{j-1})) \tag{2}$$

where S is the SE block and F is the channel-fuse module. Each basic sub-tree includes two SE blocks and a channel-fuse module. The SE block is a little different from the classical SE block, the structure of which can be observed in Fig. 3. In our SE block, a convolution layer is first used to generate the feature map with appropriate channels. Then we replace the linear layers of 1×1 convolution layers to form a bottleneck structure.

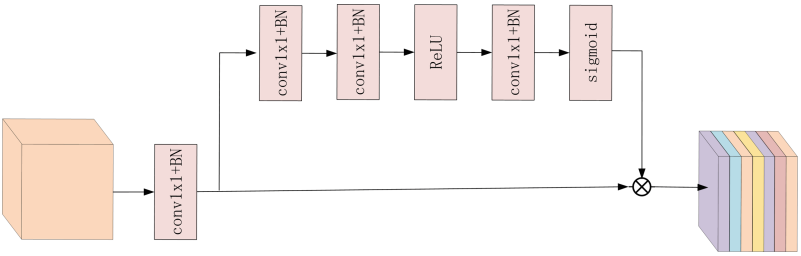


Fig. 3. The details of SE block in tree stream

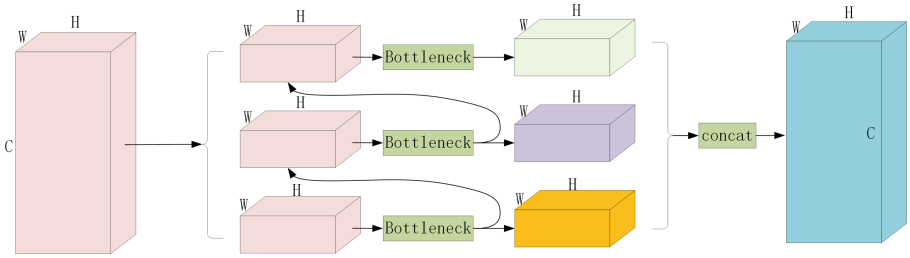


Fig. 4. The details of channel-fuse module in tree stream.

3.2 Channel-Fuse Module

Channel-fuse module is a basic module in tree stream, contributing to fuse and discover important context in the upper-level feature maps of tree stream. Many methods just use simple adding operation to fuse two input feature maps without considering the connection between channels. To solve this problem, we propose a channel-fuse module to better utilize channel context for scene parsing.

As vividly shown in Fig. 4, the input features $T_i, T_2 \in \mathbb{R}^{C\times H\times W}$ are added together, then we split the features along the channels and produce several small features denoted as t_i, where $t_i \in \mathbb{R}^{C'\times H\times W}, i \in 1, ..., N$. C' is the channel number of these feature maps after split and N is the number of small feature maps. After obtaining feature maps t_i, we further generate the feature maps $f_i, i \in 1, ..., N$ via the channel-fuse operation which is given as follows:

$$f_i = \begin{cases} B_i t_i & \mathrm{i} = 1 \\ B_i(f_{i-1} + t_i) & \mathrm{i} = 2, ..., \mathrm{N} \end{cases} \tag{3}$$

$$F = \Omega_{i=1}^{N} f_i \tag{4}$$

In channel-fuse operation, we first produce the first feature map $f_1 \in \mathbb{R}^{C'\times H\times W}$ via a bottleneck block. The bottleneck block, denoted as B_i, includes two 3×3 convolution to decrease to the input channel C' to $\frac{1}{4}C'$ and a convolution layer with 1×1 filters to increase the channel $\frac{1}{4}C'$ to C'. We denote the bottleneck block as $B_i, i \in 1, ..., N$. While for other feature maps, we add f_{i-1} and t_i. Finally, we apply concatenation operation Ω to f_i and generate the feature map $F \in \mathbb{R}^{C\times H\times W}$, enhanced by our channel-fuse module.

4 Experiment

We conduct comprehensive experiments on urban scene understanding dataset Cityscapes [6]. Experimental results demonstrate our TCFNet has the ability to make dense prediction and achieve remarkable performance on Cityscapes.

4.1 Dataset

Cityscapes [6] dataset is tasked for scene understanding, which provides 5,000 finely annotated images and 20,000 coarsely annotated images captured from 50 different cities. The finely annotated 5,000 images contain 2,975, 500, and 1,525 images for training, validation, and testing, respectively. The labeled images of the test dataset are not given. So to verify the performance on the test dataset, the images generated by the model need to be submitted to the official server. Each image in this dataset holds the resolution of 1024×2048 and contains 19 classes for scene parsing.

4.2 Implement Details

Backbone. We use ResNet-101 pretrained on the ImageNet dataset as the backbone of ResNet Stream, and replace the convolutions within the last two blocks by dilated convolutions with dilation rates of 2 and 4, following PSPNet [34]. We save feature maps generated by stage 1, 2, 3, 4, with the stride 4, 8, 8, 8 to respectively feed them into tree stream. Meanwhile, we replace the standard BatchNorm with InPlace-ABN [21] to the mean and standard-deviation of BatchNorm across multiple GPUs.

Tree Stream. We employ a dimension reduction module (1×1 convolution) to reduce the channels of the feature maps output from ResNet stream by half. To gain a high resolution feature map, we keep the resolution of the feature map on the leftmost branch in tree stream without any down-sampling operation. The SE block in our tree stream adjusts the channels of the feature maps of the left and right branch to the same in the basic sub-tree unit of tree stream, and refines them to have better representations.

Loss Function. We employ class-balanced cross entropy loss with the same weight in OCNet [31] on both the final output of tree stream and mediate output from ResNet stream. We use the feature map output from tree stream as the final output, where the weight over the final loss is 1 and the auxiliary loss is 0.4 following settings proposed in PSPNet [34].

Training Settings. We adopt stochastic gradient descent with mini-batch for training. Following prior work [31,34], the initial learning rate is set as 0.01 and the weight decay is 0.0005 by default. The input image size is 1024×2048, and we employ crop size as 769×769. We only use 2975 train-fine images to train our model with the batch size of 8. We do all experiments with $2 \times$ SKU200 GPUs. We train our model for about 40 K iterations, taking about 41 h. Following [2,3], we apply the "poly" learning rate strategy in which the initial rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$.

4.3 Comparison with State-of-the-Art

We compare our TCFNet with other state-of-the-art semantic segmentation solutions on Cityscapes validation set. As shown in Table 1, DeepLabv3 [3], OCR [30] and CCNet [10] all use ResNet-101 as the backbone. While DeepLabv3+ [4] and DPC [1] take stronger backbones like Xception-65 and so on, DeepLabv3 [3] and CCNet [10] take the structure of multiple branches to handle the multi-scale problem. These methods only take the final output of the backbone network to discover multi-scale context information. We also discover the multiple context information in the multiple stage feature maps. The result shows that our proposed TCFNet achieves the state-of-art performance. In order to view the effectiveness of our model more intuitively, we visualize the final output of our TCFNet on Cityscapes validation set, shown in Fig. 5.

In addition, we also train our TCFNet on the training and validation datasets with the pretrained ResNet-101 as the backbone, following [28,34] do. We evaluate our best learned TCFNet on the test set by submitting our test results to the official server of Cityscapes. Most of the methods like [2,28,32,34] take ResNet-101 as backbone and we take some representative methods as comparison. From Table 2, it is obvious that our method outperforms all the previous techniques in Table 2 in Cityscapes test dataset. Among the methods, BiSeNet [28] takes the U-shape in the context path and widely applies the SE block. While the tree-structure taken in TCFNet generates the feature maps layer by layer to

avoid the negative effects of low-level features. Result shows that our TCFNet outperforms BiSeNet [28] by nearly 1.0%.

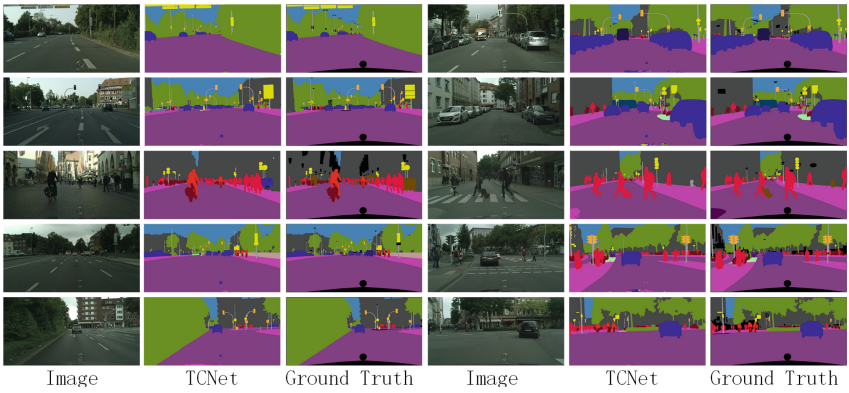


Fig. 5. Visualization of the predicted images compared to ground truth on Cityscapes validation set.

Table 1. Comparison with state-of-the-arts on Cityscapes validation set.

Methods	Backbone	mIoU(%)	
		w/o multi-scale	w/multi-scale
DeepLabV3 [3]	ResNet-101	77.8	79.3
DeepLabV3+ [4]	Xception-65	79.1	–
DPC [1]	Xception-71	80.8	–
FCN + OCR [30]	ResNet-101	79.6	80.6
CCNet [10]	ResNet-101	–	81.3
FCN(baseline)	ResNet-101	73.5	75.8
TCFNet(ours)	ResNet-101	**81.6**	**81.9**

4.4 Ablation Study

To further confirm the effectiveness of the TCFNet, we adopt some ablation experiments on the test set of Cityscapes with different settings. We utilize online hard example mining (OHEM), multi-scale (Ms), left-right flipping (Flip) and training with a validation set (w/ Val) to observe the performance of TCFNet on Cityscapes test dataset. All related results are presented in Table 3.

Table 2. Comparison with leading competitive models on Cityscapes test set.

Methods	Validation	Backbone	mIoU(%)
FCN 8s [18]	×	VGG16	63.1
DeepLab-v2 + CRF [2]	×	ResNet101	70.4
SAC [32]	×	ResNet101	78.1
PSPNet [34]	×	ResNet101	78.4
PSANet [35]	×	ResNet101	78.6
Adelaide [16]	✓	VGG16	66.4
RefineNet [15]	✓	ResNet101	73.6
DUC-HDC [26]	✓	ResNet101	77.6
DSSPN [14]	✓	ResNet101	77.8
BiSeNet [28]	✓	ResNet101	78.9
TCFNet(ours)	×	ResNet101	**78.7**
TCFNet(ours)	✓	ResNet101	**79.7**

OHEM. OHEM [22] is used to select some hard examples as training samples to improve the network performance and hard examples refer to samples with diversity and high loss. We choose the same setting in [41] on Cityscapes and improve 0.3 on test dataset.

Training W/Validation Set. We further boost the performance of our TCFNet on the test set by adding the validation set for training. We keep the training setting except adding another 20 K training iterators and improves the performance from 79.0 to 79.5.

Ms + Flip. We also adopt left-right flip and multi-scales including [0.75×, 1×, 1.25×] to gain a better result from 79.5 to 79.6.

Fine-Tuning. Following DeepLabv3 [3], we fine-tune our model by training our model for additional 10 K iterations with fine-labeled dataset. This also improves performance from 79.6 to 79.7.

5 Conclusion and Future Task

In this paper, we proposed TCFNet for scene parsing, in which the tree-like structure and the channel-fuse module were designed together to capture the context information from different stage feature maps with various resolutions and channels. We demonstrated our TCFNet's effectiveness on the challenging scene parsing benchmark like Cityscapes and visualized multiple examples. In our future task, we will apply TCFNet to other challenging tasks like object detection and so on.

Table 3. The effect of the OHEM, MS + Flip, training w/ the validation set and fine-tuning, the results are tested on Cityscapes test set.

OHEM	w/Val	Ms + Flip	Fine-tuning	Test. mIoU(%)
×	×	×	×	78.7
✓	×	×	×	79
✓	✓	×	×	79.5
✓	✓	✓	×	79.6
✓	✓	✓	✓	**79.7**

Acknowledgment. This work was supported by National Natural Science Foundation of China (Grant 61303029), Fundamental Research Funds for the Central Universities of China (Grant 191010001), Foundation of Hubei Key Laboratory of Transportation Internet of Things (Grant 2018IOT003), and Hubei Provincial Natural Science Foundation of China (Grant 2017CFA012).

References

1. Chen, L.C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In: Advances in Neural Information Processing Systems, pp. 8699–8710 (2018)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
3. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking Atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with Atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp. 801–818 (2018)
5. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781 (2016)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
7. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)

10. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: CCNet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 603–612 (2019)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. Liang, X., Zhou, H., Xing, E.: Dynamic-structured semantic propagation network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 752–761 (2018)
15. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925–1934 (2017)
16. Lin, G., Shen, C., Van Den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3194–3203 (2016)
17. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: Looking wider to see better. arXiv preprint arXiv:1506.04579 (2015)
18. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
19. Ma, B., Entezari, A.: Volumetric feature-based classification and visibility analysis for transfer function design. IEEE Trans. Visual Comput. Graphics **24**(12), 3253–3267 (2017)
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241. Springer (2015)
21. Rota Bulò, S., Porzi, L., Kontschieder, P.: In-place activated BatchNorm for memory-optimized training of DNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5639–5647 (2018)
22. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)
23. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. arXiv preprint arXiv:1902.09212 (2019)
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
25. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
26. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.: Understanding convolution for semantic segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1451–1460. IEEE (2018)
27. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)

28. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: BiseNet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 325–341 (2018)
29. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1857–1866 (2018)
30. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. arXiv preprint arXiv:1909.11065 (2019)
31. Yuan, Y., Wang, J.: OCNet: object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)
32. Zhang, R., Tang, S., Zhang, Y., Li, J., Yan, S.: Scale-adaptive convolutions for scene parsing. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2031–2039 (2017)
33. Zhang, X., Zhou, X., Lin, M., Sun, J.: ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
34. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881–2890 (2017)
35. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: PsaNet: point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 267–283 (2018)
36. Zhong, X., Gong, O., Huang, W., Li, L., Xia, H.: Squeeze-and-excitation wide residual networks in image classification. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 395–399. IEEE (2019)

Detecting Cues of Driver Fatigue on Facial Appearance

Ann Nosseir[1,2(✉)] and Mohamed Esmat El-sayed[1]

[1] British University Egypt, Cairo, Egypt
{Ann.nosseir,mohamed126133}@bue.edu.eg, nosseir12@yahoo.co.uk
[2] Institute of National Planning, Cairo, Egypt

Abstract. Driver fatigue causes tragic events and hazardous consequences in transportation systems. Especially, in developing countries, drivers have longer working hours and drive longer distances with short breaks to gain more money. This paper develops a new real time, low cost, and non-intrusive system that detects the features of fatigue drivers. It detects cues from the face of people who didn't sleep the right hours or who have over worked. It identifies eye-related cues such as red eyes and skin-related cues who have like dark areas under the eye. This work uses CascadeObjectDetector that supports the Haar Cascade Classifier, Local Binary Patterns (LBP) and Histograms of Oriented Gradients (HOG) to develop a new algorithm, and locates the areas under the eye and reference areas on the face to compare the skin color tone. It uses the semi-supervised anomaly detection algorithm to recognize abnormality of the area under the eyes and eye redness. The system was evaluated with 7 participants with different skin colors and various light conditions. The results are very promising. The accuracy is quite high. All cues are detected correctly.

Keywords: Driver fatigue · Image processing · Facial cues

1 Introduction

Driver fatigue is a serious threat to road safety. Insufficient sleep and a long stint at work increases the risk of road accidents. Detection of driver fatigue or drowsiness in the early stages can help in reducing this risk. Automotive industry has realized the importance of automatic detection of driver fatigue. It is taking priority in companies such as Toyota [1], Volkswagen [2] and Nissan [3]. Toyota works on detecting car lane deviation. Nissan's Maxima model tracks the driver's steering patterns and identifies unusual deviation from the regular pattern [4]. Volkswagen [2] offers a lane tracking system by tracking pedal use and erratic steering wheel movements to judge driver fatigue level.

Research reported in [5, 6] is exploring other methods using camera to detect cues of fatigue from the face. These cues are limited to eye blinking, yawning, and head pose. Sundelin et al. [7] investigated other facial cues to recognize sleep deprivation. They have classified cues into eye-related cues such as red eyes and skin-related cues like darker circles under the eyes.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 710–723, 2021.
https://doi.org/10.1007/978-3-030-55180-3_54

This paper presents a novel work that automatically detects skin and eye-related cues such as dark circles under the eyes and red eyes. This non-intrusive approach utilizes image processing and anomaly detection machine learning techniques.

In data mining, anomaly detection is referred to the identification of items or events that do not conform to an expected pattern or to other items present in a dataset. Typically, these anomalous items have the potential of getting translated into some kind of problem [8].

This paper starts with the related work done to solve the problem and the current solutions of detecting fatigue drivers. This is followed by the proposed system and its evaluation in different conditions. The paper ends with a discussion and conclusion.

2 Related Work

2.1 Current Systems

Research in the area of fatigue detection uses either intrusive or non-intrusive devices or a hybrid of both.

The former, i.e. the intrusive uses sensors attached to the human body to detect physiological fatigue signals from breathing rate, heart rate, brain activity, muscles, body temperature and eye movement. The physiological signals start to change in early stages of drowsiness [9, 10].

Devices such as Electroencephalogram EEG, Electrocardiogram ECG, and Electroculogram EOG are attached to the body to read these signals. Electrocardiogram (ECG) signals of the heart vary significantly between the different stages of drowsiness [11–13]. Electroencephalography (EEG) signals of the brain are categorized as delta, theta, alpha, beta and gamma. A decrease in the alpha frequency band and an increase in the theta frequency band indicate drowsiness [13].

Sweat Rate and Electromyogram (EMG) [14] are detected to discriminate mental stress and stress caused by physical discomfort; like the one that could occur during abrupt acceleration or deceleration in a car race. Additionally, Skin Potential Response (SPR) SPR [14] measures the nervous pulses which activate sweat glands.

Electrooculography (EOG) detects the electric difference between the cornea and the retina that reflects the orientation of the eyes. It identifies the Rapid Eye Movements (REM) which occur when a subject is in a drowsy state [15]. These systems need to be attached to the human body to get any reading and this is main challenge to implementing these systems. They cause discomfort to the drivers especially if they are attached for a long time.

The latter i.e., non-intrusive, uses devices or sensors attached to the car. For example, a camera captures eyes' closed [16, 17], head nodding, orientation [18], and yawning [19].

These techniques are tested with images of faces that have different states of fatigue or with videos and a few are tested with real time streaming videos. The evaluations as well have been done in a lab with a simulator. A few have been tested outdoors.

To gain more confident in the techniques, the accuracy of these techniques need to be tested in different conditions such as in different lighting conditions of day or night.

These conditions affect the accuracy of extracting the facial features and the measures of eye blinks.

This work presented in this work develops a system that detects the face features and extracts a fatigue measures and tests this system in different light conditions.

3 Methodology

This novel work uses non-intrusive devices and detects cues of the drivers' fatigue. It develops an original images processing algorithm to locate the area under the eye and a reference area on the face to compare the skin color tone.

Additionally, it uses the anomaly detection semi-supervised machine learning technique to spot fatigue cues. The anomaly detection, known as outlier detection "is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data" [8]. In this work, the system applies a semi-supervised learning algorithm where it is trained on a combination of labeled and unlabeled data and uses a statistical scheme for the anomaly detection. Here are the main steps of the system (see Fig. 1).

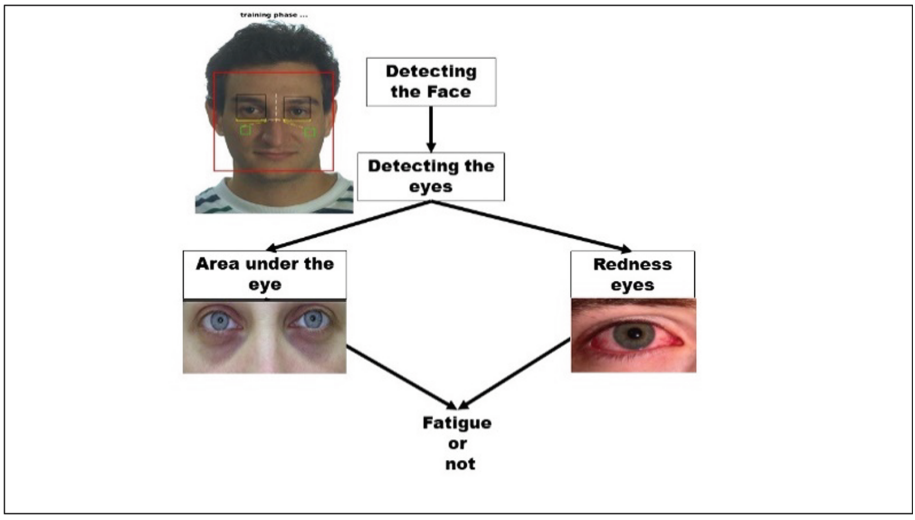


Fig. 1. The algorithm steps

1. Image Acquisition
2. Image Processing and Detecting Face Areas
    1. Detect Face.
    2. Detect Eyes.
    3. Detect Areas Under the Eye.

4. Detect Reference Areas.
5. Calculate the RGB in Each Area.

3. Anomaly Detection

1. Detect the Anomalies of the Area Under Eyes.
2. Detect Unusual Red Eyes.

3.1 Image Acquisition

To develop a model that detects the fatigue, the system uses images from the National Institute of Standard and Technology dataset [20]. These images are 512 pixels in width and 768 pixels in height. The images include people with different nationalities and with different skin color tones. For the training model, the system used 86 images that includes 51 images for male and 35 images for female.

3.2 Image Processing and Detecting Face Areas

3.2.1 Face Detection

To detect the face and the eyes regions, this work used a combination of different algorithms, namely, Haar Cascade Classifier, Local Binary Patterns (LBP), and Histograms of Oriented Gradients (HOG) [21].

The Haar Cascade Classifier is an object detection method. It uses the Viola-Jones algorithm to detect people's faces, noses, eyes, mouth, or upper body. It was proposed in 2001 by Paul Viola and Michael Jones [22]. Haar Cascade Classifier overlays the positive image over a set of negative images to train the classifier. A Haar-like feature considers neighboring rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image.

The Local Binary Pattern (LBP) [23] is an effective texture descriptor for images which thresholds the neighboring pixels based on the value of the current pixel. It has discriminative power and computational simplicity. The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. Histogram of oriented gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image detection window, or region of interest (ROI) [21].

The CascadeObjectDetector library supports three types of features: Haar Cascade Classifier, Local Binary Patterns (LBP), and Histograms of Oriented Gradients (HOG) algorithms [24]. In Fig. 2, the red i.e., largest rectangle represents the boundaries of the face.

3.2.2 Eyes Detection

Identifying the face region increases the ability of detecting each eye individually with high accuracy. Within the face area in the image, the same algorithms are used for

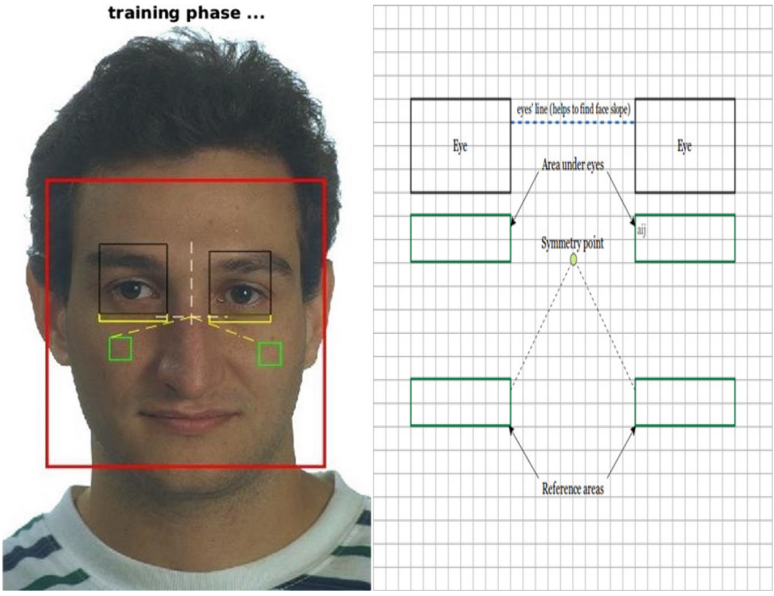


Fig. 2. Locating the face, eyes, under the eyes and the reference areas

detecting the eyes and the area under the eye. In Fig. 2, two intermediate sized rectangles that represent the boundaries of the eyes.

3.2.3 Areas Under the Eye Detection

After identifying the eyes regions, the system detects the regions under the right and left eyes. It uses the location of eyes to get the area under eyes. The system adds a height to the area under the eyes equal to 10 pixels. It calculates its starting point with respect to the eye boundaries. Figure 3 displays the position x, y of the eye rectangle and the d_x and d_y of the area under the eye rectangle.

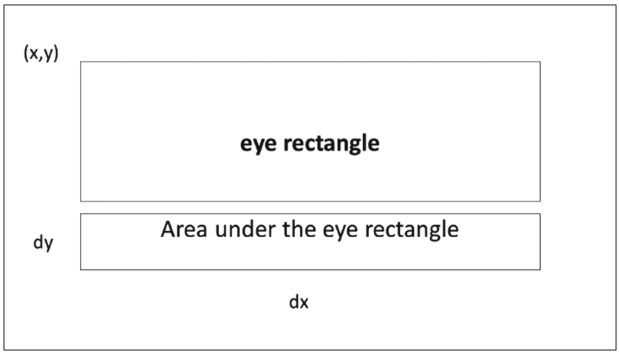


Fig. 3. The eyes and area under the under rectangles

For each eye, labelling the x axes and y axes are done as it is labeled in Fig. 3.

For the left eye, d_{xl} represents the width values of the area under the left eye while d_{yl} represents the height of the rectangle. The d_{yl} values range from 0 to 10 pixels. The area under the left eye is $d_{xl} \times d_{yl}$.

For the right area, the under the eye is represented by $d_{xr} \times d_{yr}$. where d_{xr} represents the width of the area eye and d_{yr} represents the height of the area. d_{yr} ranges from 0 to 10 pixels. In Fig. 2, the rectangles under the eyes are the boundaries of these areas.

3.2.4 Reference Area

To decide whether the person has dark circles under her/his eyes or not, the system identifies a reference area on the face to compare between the skin tone of this area and the area under the eyes. From the area under the eyes, the system finds a symmetry point to draw a vertical line. This point is an intersection of two white dashed lines in Fig. 2:

- To draw the vertical symmetric line, the distance between the rectangle of both eyes is calculated and divided by 2. This is to get the central point between the left eye and the right eye.
- Using the rectangles bounding each eye, the system draws a horizontal line that links the bottom lines of the rectangles with a length of 30 pixels.
- Then, the system draws two perpendiculars lines from the intersect point (symmetric point) of the vertical and horizontal symmetric lines. To draw these lines and the reference area accurately, the system calculates the offset because the 90° angle of the person in the image is not the same for all images. The offset rule is:

$$\emptyset = \tan^{-1} \frac{yi - yr}{xi - xr} \tag{1}$$

- Knowing the calculated angle, the system draws a line of 30 pixels long to get the start points of each reference areas. The values of the height and width are fix to 10 pixels.

The reference areas are shown in Fig. 2.

3.2.5 Calculate the RGB and Get the Vector of Each Area

To compare both areas color tones, i.e., the areas under the eye and the reference areas, the system extracts the color components of red, green, and blue. Each pixel of i^{th} row and j^{th} column has a_{ij} which is a vector of the three color components.

$$a_{ij} = (r \cdot g \cdot b)_{ij} \tag{2}$$

Figure 4, 5 and 6 are a visualization of each color, i.e. red, green and blue components, of the feature vector.

A_{eye} vector has the RGB values of all pixels' area under the eye. A_r vector has RGB values of all pixels in the reference area.

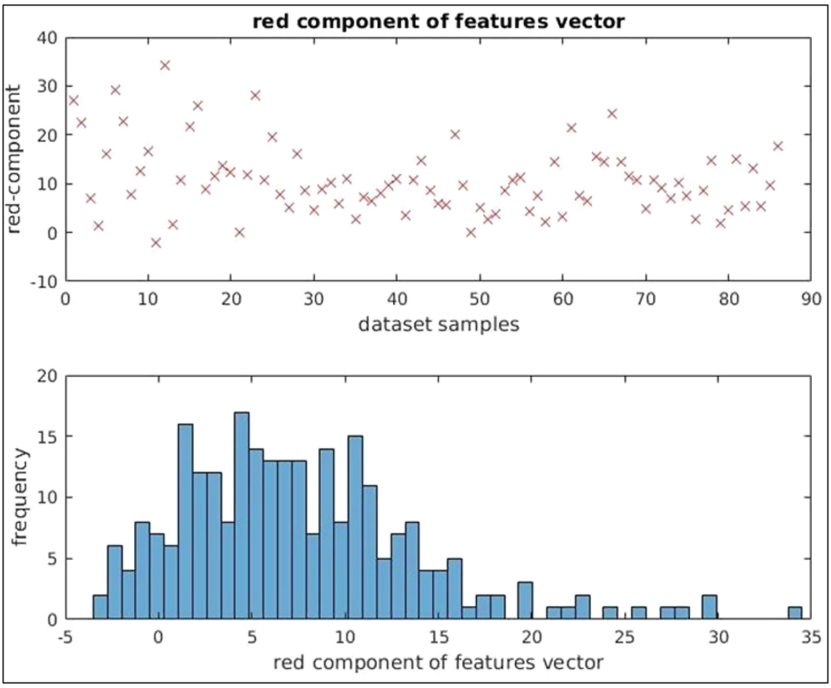


Fig. 4. The red component of the feature vector

3.3 Anomaly Detection

3.3.1 Detecting the Anomalies of the Area Under Eyes

Figure 7 shows the steps of detecting fatigue related to the skin cues. After identifying the area under the eye and reference areas, and calculating RGB, the system extracts the feature vector, calculates the Multivariate Gaussian Distribution and fits the parameters. From that, the probability density function model is generated and the threshold is decided.

3.3.1.1 Feature Vector

The feature vector λ is the subtraction of the areas under the eyes from the reference area RGB values.

$$\lambda = \bar{A}_{eyes} - \bar{A}_r \tag{3}$$

Feature vector is the difference of the RGB vectors

$$\lambda = (\lambda_r, \lambda_g, \lambda_b) \tag{4}$$

We get the matric of all 86 images

$$\text{Feat} - \text{MAT} = \begin{matrix} \lambda_r^{(1)} & \lambda_g^{(1)} & \lambda_b^{(1)} \\ \vdots & \vdots & \vdots \\ \lambda_r^{(86)} & \lambda_g^{(86)} & \lambda_b^{(86)} \end{matrix} \tag{5}$$

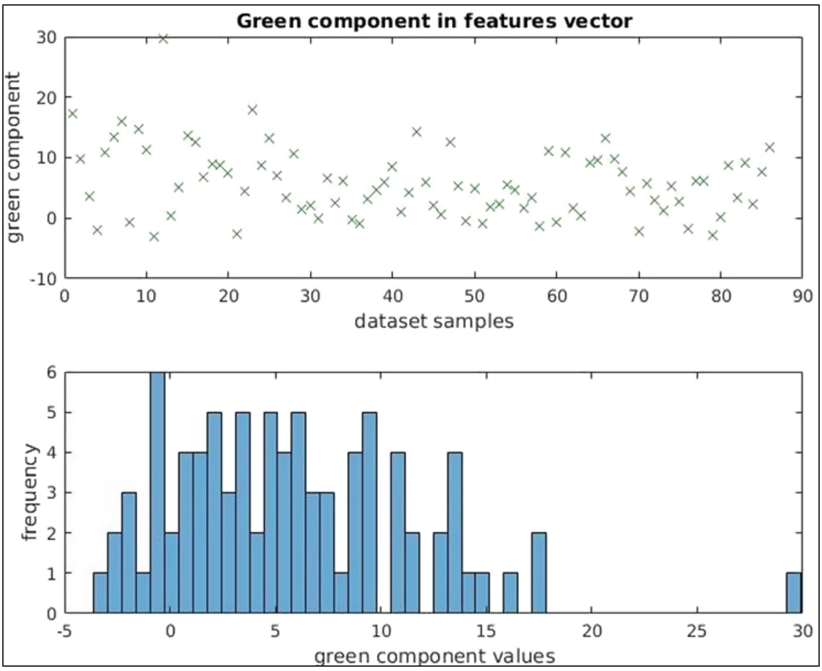


Fig. 5. The green component of the feature vector

3.3.1.2 Anomaly Detection Algorithm Using Multivariate Gaussian Distribution and Fit the Parameters

The system applies the Multivariate Gaussian Distribution on the feature vector λ that produces a vector for μ and another for Σ.

Given a training set of examples, $x^{(1)}, x^{(2)} \ldots .x^{(m)}$ where each example is a vector $x \in R$, $p(x) = \prod_{j=1}^{n} p\left(x_{j;\mu_j;\sigma_j^2}\right)$

calculate:

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \tag{6}$$

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j\right)^2 \tag{7}$$

$$\mu = \begin{matrix} \mu_r \\ \mu_g \\ \mu_b \end{matrix} \tag{8}$$

$$\sum = \begin{matrix} \sigma_r^2 & & \\ & \sigma_g^2 & \\ & & \sigma_b^2 \end{matrix} \tag{9}$$

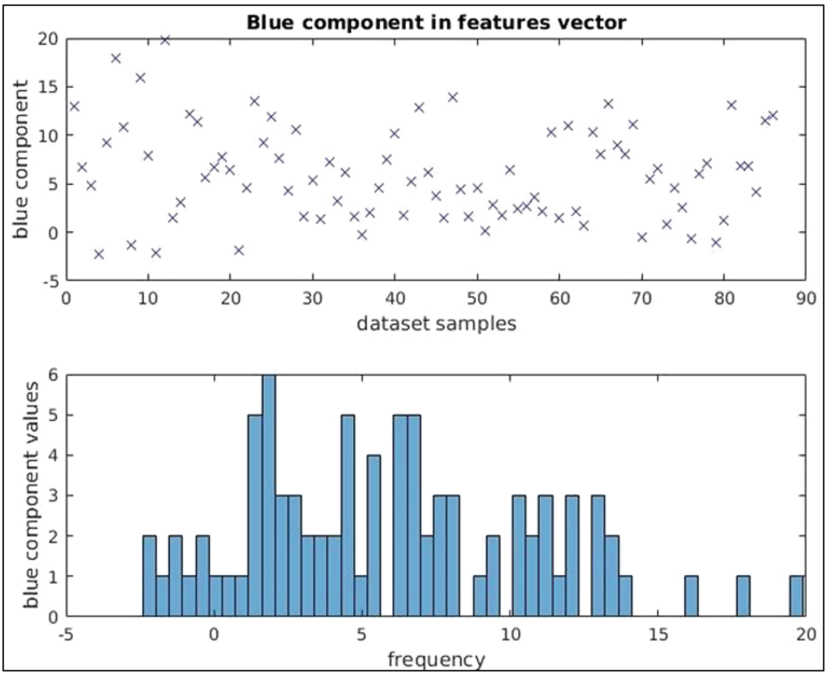


Fig. 6. The blue component of the feature vector

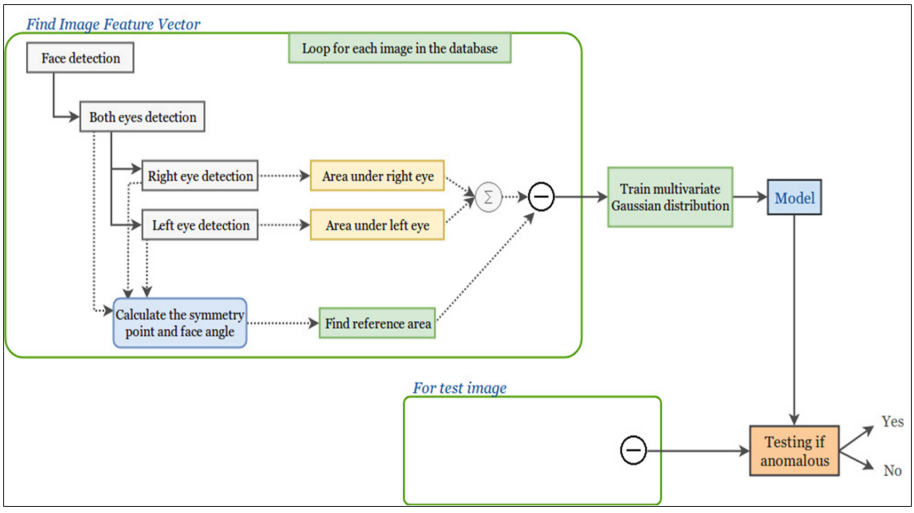


Fig. 7. Detecting the anomalies of the area under eyes.

3.3.1.3 Probability Density Function Model

The probability density function is to get the (x) random variable equation:

$$p(\lambda;\mu,\Sigma)=\left(\frac{1}{((2\pi^{n/2})|\Sigma|^{1/2})}\exp\left(-1/2(\lambda-\mu)\right)^{T\Sigma^{-1}(\lambda-\mu)}\right) \tag{10}$$

The size of vector λ; $n = 3$. After training this model, we can use the probability density function to recognize any abnormal dark areas under the eyes in any untrained image.

3.3.1.4 Anomaly Detection and Identifying the Threshold

In order to flag anomaly detection and define the threshold, we had to experiment with the images that define normality to determine the best and most realistic threshold. The threshold is empirically chosen to be 0.1x mean of the probability density function of the trained model.

The threshold is the Mean calculated which is 1.5121e.05. This sample is declared an anomaly if the probability of the new picture is less than the expected value calculated from the trained model $(x) < \Sigma$.

3.4 Detecting Unusual Red Eyes

Our second feature is to detect the unusual red eyes. To check whether the eye has unusual red color or not (see Fig. 8). The system follows these steps:

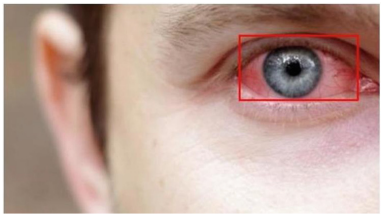


Fig. 8. Sample of the red eye area.

- Detecting the eye region as performed in the earlier steps.
- In this region, the system performs the RGB function and the regular red color distribution of the eyes is extracted
- The mean of the red value of the normal images is calculated.
- From all the images the threshold is computed and it is set to the mean +2. In this model, the threshold is 113.5549.
- To detect abnormality, the mean of the red value of the new image is compared with the threshold. If the average of the red is greater than the red threshold that means that the red has high intensity.

4 Evaluation

To gain some confident of the algorithm, this work has been evaluated with image of regular people who are tired and fatigue.

4.1 Equipment

The evaluation was carried out using a laptop, which is HP intel CORE i5 7th Gen. with a 4 GB RAM and 500 GB Hard Disk. The Laptop's Camera and External HD 1080 Camera attached to the laptop for a live video streaming works as input to the system.

4.2 The Participants and Procedures

The system was assessed with seven participants. They are three females. The age distribution is as follow. One female average age is between 20–25 and the second is between 45–50 and the third is between 65–70. There are 4 males. Average age of three of them is between 20 and 25 and one is between 30 and 35. Their color skin varied.

The system has been tested with the variance of the light shade because the shade can affect detecting the area under the eye and the reference area (see Fig. 8 and 9).

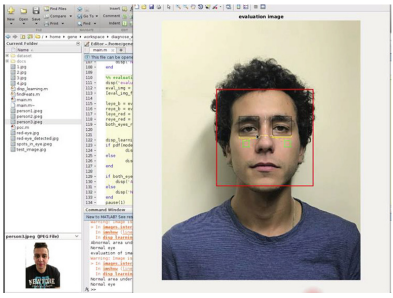


Fig. 9. Normal eye and area under the eyes.

To make sure the participants are fatigue. The study was done at night and the participants were too tired and did not sleep well in the previous night.

In the study, the participants were asked to sit in five different places to change the light and the shades and the camera will take picture of your face.

4.3 Results

The results are shown in Fig. 9 and 10. In Fig. 9, the picture is for a male and his age is between 20 and 25. His eye is normal and he doesn't have dark area under his eye. The system recognized correctly the eye and the area under the eye.

Figure 10 has pictures of the other six participants. The images have different light shade and all participants have dark areas under their eyes and the system successfully detected.

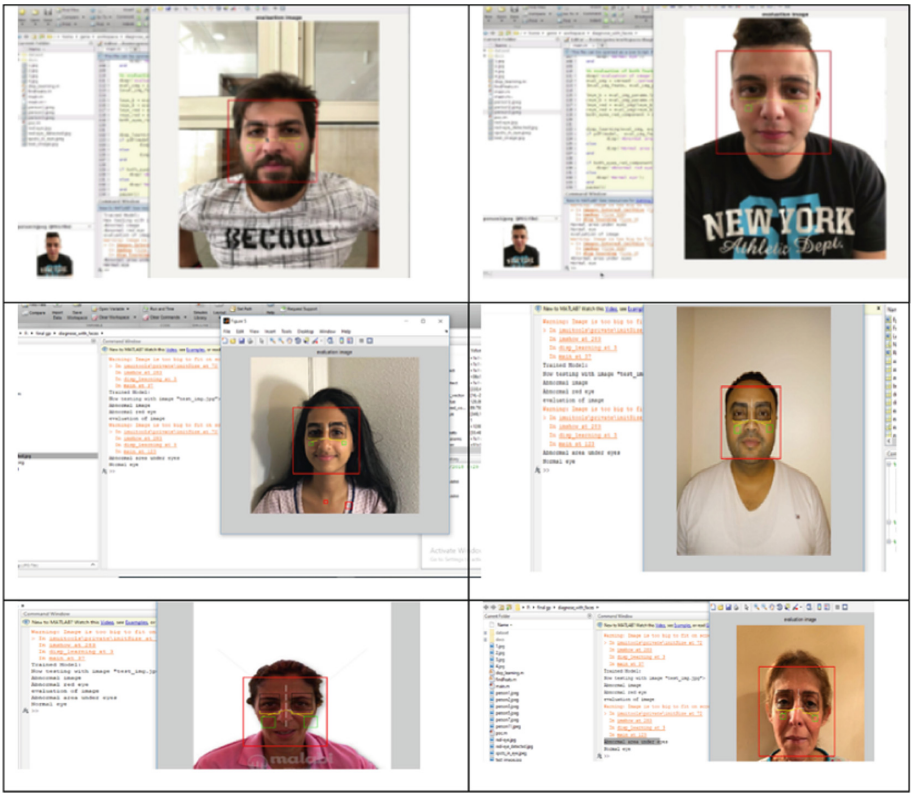


Fig. 10. Normal eye and abnormal under.

5 Discussions and Conclusions

This work has presented a novel contribution in the domain of detecting driver fatigue. The system doesn't require continuously monitoring of the face. It can be used once in a while to check the drive and send the appropriate alert.

The Viola-Jones algorithm through the CascadeObjectDetector identified the face and the eye regions. The system develops a new approach to define the dark area under the eyes and reference areas on the face to compare the skin color tone.

The system builds a semi-supervised anomaly detection model to detect the abnormality of the area under eyes and eye redness.

The system is evaluated with seven participants with different skin colors and various light conditions. The results are very promising. The accuracy is quite high. All cues are detected correctly.

References

1. Toyota: Toyota Safety Sensee. https://www.toyota-europe.com/world-of-toyota/safety/toyota-safety-sense. Accessed 28 Mar 2019
2. Volkswagen: Driver Assist. https://www.volkswagen.co.uk/technology/driver-assist. Accessed 28 Mar 2019
3. Nissan: Driver Attention Alert. http://www.nissantechnicianinfo.mobi/htmlversions/2015_June-July_Issue2/Driver_Attention_Alert.html. Accessed 28 Mar 2019
4. F.I. Release and H. I. Works: Nissan's 'Driver Attention Alert' Helps Detect Erratic Driving Caused By Drowsiness and Inattention, pp. 1–2 (2016)
5. Nosseir, A., Hamad, A., Wahdan, A.: Detecting drivers' fatigue in different conditions using real time non-intrusive system. In: Fourth International Congress on Information and Communication Technology - ICICT 2019, London, vol. 2, pp. 156–164. Springer, Singapore (2019)
6. Rastgoo, M.N., Nakisa, B., Rakotonirainy, A., Chandran, V., Tjondronegoro, D.: A critical review of proactive detection of driver stress levels based on multimodal measurements. ACM Comput. Surv. **51**(5), 1–35 (2018)
7. Sundelin, T., Lekander, M., Kecklund, G., Van Someren, E.J.W., Olsson, A., Axelsson, J.: Cues of fatigue: effects of sleep deprivation on facial appearance. Sleep **36**(9), 1355–1360 (2013)
8. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 85–126 (2004)
9. Chowdhury, A., Shankaran, R., Kavakli, M., Haque, M.M.: Sensor applications and physiological features in drivers' drowsiness detection: a review. IEEE Sens. J. **18**(8), 3055–3067 (2018)
10. Chen, L., Zhao, Y., Zhang, J., Zou, J.: Automatic detection of alertness/drowsiness from physiological signals using wavelet-based nonlinear features and machine learning. Expert Syst. Appl. **42**, 7344–7355 (2015)
11. Chen, J., Wang, H., Hua, C.: Assessment of driver drowsiness using electroencephalogram signals based on multiple functional brain networks. Int. J. Psychophysiol. **133**(July), 120–130 (2018)
12. Wang, P., Min, J., Hu, J.: Ensemble classifier for driver's fatigue detection based on a single EEG channel. IET Intell. Transp. Syst. **12**(10), 1322–1328 (2018)
13. Wang, F., Wang, H., Fu, R.: Real-time ECG-based detection of fatigue driving using sample entropy. Entropy **20**(3), 196 (2018)
14. Affanni, A., Bernardini, R., Piras, A., Rinaldo, R., Zontone, P.: Driver's stress detection using skin potential response signals. Meas. J. Int. Meas. Confed. **122**, 264–274 (2018)
15. Papadelis, C., Chen, Z., Kourtidou-papadeli, C.: Monitoring sleepiness with on-board electrophysiological recordings for preventing sleep-deprived traffic accidents. Clin. Neurophysiol. **118**, 1906–1922 (2007)
16. Magaña, V.C., Organero, M.M., Álvarez-García, J.A., Rodríguez, J.Y.F.: Estimation of the optimum speed to minimize the driver stress based on the previous behavior. In: Advances in Intelligent Systems and Computing, vol. 476, pp. 31–39 (2016)
17. Ramodhine, K., Panchoo, S.: Emerging trends in electrical, electronic and communications engineering. In: Emerging Trends in Electrical, Electronic and Communications Engineering, Lecture Notes in Electrical Engineering, vol. 416 (2017)
18. Wathiq, O., Ambudkar, B.D.: Optimized driver safety through driver fatigue detection methods. In: International Conference on Trends in Electronics and Informatics ICEI 2017, pp. 68–73 (2017)

19. Abtahi, S., Hariri, B., Shirmohammadi, S.: Driver Drowsiness Monitoring Based on Yawning Detection, July 2015
20. Face Recognition Technology (FERET). https://www.nist.gov/programs-projects/face-recognition-technology-feret
21. Dalal, N. et al.: Histograms of Oriented Gradients for Human Detection To cite this version: HAL Id: inria-00548512 Histograms of Oriented Gradients for Human Detection (2010)
22. Viola, P., Way, O.M., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
23. He, D.-C., Wang, L.I.: Texture unit, texture spectrum, and texture analysis. IEEE Trans. Geosci. Remote Sens. **28**(4), 509–512 (1990)
24. Train a Cascade Object Detector. https://uk.mathworks.com/help/vision/ug/train-a-cascade-object-detector.html

Discriminative Context-Aware Correlation Filter Network for Visual Tracking

Xinjie Wang[1], Weibin Liu[1(✉)], and Weiwei Xing[2]

[1] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
wbliu@bjtu.edu.cn

[2] School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China

Abstract. In recent years, discriminative correlation filter (DCF) based trackers using convolutional features have received great attention due to their accuracy in online object tracking. However, the convolutional features of these DCF trackers are mostly extracted from convolutional networks trained for other vision tasks like object detection, which may limit the tracking performance. Moreover, under the challenge of fast motion and motion blur, the tracking performance usually decreases due to the lack of context information. In this paper, we present an end-to-end trainable discriminative context-aware correlation filter network, namely DCACFNet, which integrates context-aware correlation filter (CACF) into the fully-convolutional Siamese network. Firstly, the CACF is modeled as a differentiable layer in the DCACFNet architecture, which can back-propagate the localization error to the convolutional layers. Then, a novel channel attention module is embedded into the DCACFNet architecture to improve the target adaption of the whole network. Finally, this paper proposes a novel high-confidence update strategy to avoid the model corruption under the challenging of occlusion and out-of-view. Extensive experimental evaluations on two tracking benchmarks, OTB-2013 and OTB-2015, demonstrate that the proposed DCACFNet achieves the competitive tracking performance.

Keywords: Correlation filter · Siamese network · Channel attention · Visual tracking

1 Introduction

With the rapid development of many visual applications such as human-computer interaction, pose estimation and autonomous driving, visual object tracking, one of the basic research tasks of computer vision, has attracted more and more attention from researchers. The key point of object tracking is how to estimate the state of an arbitrary target accurately in continuous video sequences [1–3]. Different from object detection, the tracking target object can be of any class. Despite significant advances in recent years [4–9], the performance of object tracking usually drops in many complex scenes.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 724–736, 2021.
https://doi.org/10.1007/978-3-030-55180-3_55

According to their principles, almost all the visual object trackers can be divided into two main paradigms. One is generative trackers[10–12], which predict the location and the scale of the tracking object that is best match with the target in help of the joint probability density between targets and search candidates. The other is discriminative trackers that utilize a pre-trained classifier to efficiently identify the tracking targets from a search area [13–16]. In terms of current research, discriminative trackers are more popular than generative trackers.

In recent years, as one of the discriminative trackers, discriminative correlation filter based trackers have attracted great interest by reasons of their trade-off between accuracy and speed, such as KCF [16], CN [17], DSST [18], SRDCF [19], Staple [20], AT [21], CACF [22]. Most of these DCF trackers exploit handcrafted multi-channel features (e.g., HoGs), which has a negative effect on their accuracy. Later, as the convolutional neural network (CNN) has achieved great success in object classification and object detection, the visual tracking community has paid significant attention to the combination of convolutional neural network and correlation filter tracking [23–27]. Representative DCF trackers using deep CNN features include HCF [23], DeepSRDCF [26] and UPDT [27]. However, these DCF trackers using pre-trained CNN features independently combine CNN with DCF tracking process, which can hardly benefit from deep integration of the two methods.

Recently, the CFNet [28] and the DCFNet [29] were proposed to address the deep integration of CNN and DCF tracking. These two methods unify the DCF and the fully-convolutional Siamese framework, which interpret the back-propagate of the DCF and embed the DCF into the CNN as a differentiable layer. It turns out that using CNN features that are best suited for the DCF tracker can extremely increase the tracking performance. However, the performance of these end-to-end DCF trackers often drops significantly under the challenge of fast motion and motion blur due to the boundary effects. Furthermore, these DCF trackers do not take full advantage of the context information around the object, which limits the discriminative capability of these trackers.

In addition, the CFNet and the DCFNet are not able to distinguish different tracking object because they don't conduct online adaption during online tracking. In other words, they treats tracking object equally. In particularly, their target adaption ability needs to be improved. And they usually update the model at each frame, which ignores the accuracy of the tracking results. This model update strategy is not wise when the target's tracking results are poor under some complex scene, which causes in a deterministic failure [30].

To address the aforementioned limitations, the main contributions of this paper are summarized as follows. With the goal of improving the discriminative ability, an end-to-end context-aware correlation tracking framework using CNN features is proposed, which treats CACF as a special layer to be embedded into the Siamese framework. Then, on the template branch of the DCACFNet architecture, a novel channel attention module is added after the convolutional layer to select important convolutional feature channels for different visual objects. Finally, in order to avoid the model corruption, this paper proposes a novel model update strategy that takes fully advantage of the tracking results to decide whether to update the model or not. Comprehensive experiments on OTB-2013 and OTB-2015 show the effectiveness of the proposed DCACFNet.

2 Discriminative Context-Aware Correlation Filter Network

The overall DCACFNet architecture is shown in Fig. 1. Different from SiamFC, the proposed DCACFNet architecture is asymmetric. In contrast to the search branch, the temple branch adds the channel attention module [31] after the convolutional feature transform. The outputs of these branches are fed into the CACF layer to locate the tracking target.

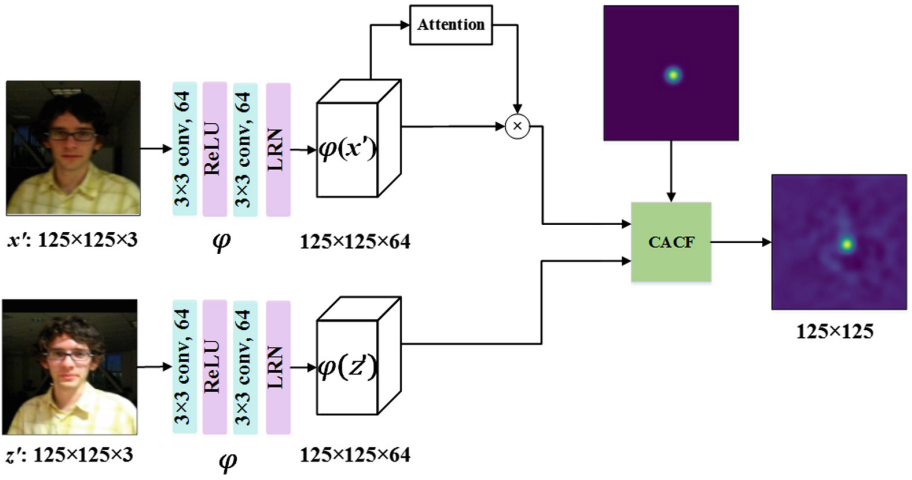


Fig. 1. The overall DCACFNet architecture.

In order to detail the DCACFNet architecture, this section briefly reviews the context-aware correlation filter at first. We subsequently drive the back-propagation of the context-aware correlation filter layer, which conducts online learning during the forward propagation. Then, a novel channel attention module is introduced to be embedded in the DCACFNet architecture. At last, the high-confidence update strategy is given to improve the tracking performance under the challenging of occlusion.

2.1 Context-Aware Correlation Filter

We start to revisit the overview of the general DCF tracker. A DCF is used for inferring the location of the target in successive frames. A DCF can be efficiently learned with the help of dense sampling around the target. In order to form the feature matrix X_0 with the circulant structure, the DCF concatenates the CNN features of which are from all possible translations in the search window of the tracking object. The solution of the DCF filter can be viewed as the optimization of the ridge regression problem.

$$\min_{w} \|X_0 w - y\|_2^2 + \lambda_1 \|w\|_2^2 \tag{1}$$

Here, the vector w denotes the learned DCF, each row of the feature matrix X_0 is composed of the features extracted from the image patch $x_0^{'}$ and its certain cyclic

shift. The two-dimensional vectorized Gaussian image is denoted by the regression objective y.

In the Fourier domain, this circulant feature matrix provides an efficient closed-form solution to the ridge regression problem of the DCF:

$$\hat{w} = \frac{\hat{x}_0^* \odot \hat{y}}{\hat{x}_0^* \odot \hat{x}_0 + \lambda_1} \tag{2}$$

Motivated by the CACF method [22], we add the global context to our correlation filter for larger discriminative power. The context sampling strategy usually plays an essential role in the tracking performance. Our CACF selects context sample patches uniformly in the surrounding of the tracking target. According to the sampling strategy, k context image patches x_i' are sampled around the target image patch x_0' of each frame. Intuitively, various target distractors and diverse background clutter are reflected in the context patches, which may be a kind of hard negative samples. Then, what we want to learn is a correlation filter that has different response for the target patch and context patches. Specifically, while its response is high for the target patch, its response is close to zero for context patches, which can suppress various target distractors.

$$\min_w \|X_0 w - y\|_2^2 + \lambda_1 \|w\|_2^2 + \lambda_2 \sum_{i=1}^{k} \|X_i w\|_2^2 \tag{3}$$

Here, both X_i and X_0 are their corresponding circulant feature matrix based on the extracted CNN features.

Similar to the solution of the DCF in the Fourier domain, the closed-form solution for our CACF can be written as:

$$\hat{w} = \frac{\hat{x}_0^* \odot \hat{y}}{\hat{x}_0^* \odot \hat{x}_0 + \lambda_1 + \lambda_2 \sum_{i=1}^{k} \hat{x}_i^* \odot \hat{x}_i} \tag{4}$$

Here, x_0 represents the CNN feature of the image patch $x_0^{'}$, i.e., $x_0 = \varphi\left(x_0^{'}\right)$, $\varphi(\cdot)$ means a feature transformation mapping of the convolutional layers in our network, the discrete Fourier transform of x_0 is denoted by the hat $\hat{x}_0$, $\hat{x}_0^*$ denotes the complex conjugate of the hat $\hat{x}_0$, and $\odot$ means the Hadamard product.

2.2 DCACFNet Derivation: Back-Propagation

Compared to CACF that performs correlation analysis using hand-crafted features, we design a novel end-to-end trainable Siamese network that learn fine-grained representations suitable for a CACF by modeling CACF as a differentiable CACF layer after convolutional layer. Due to the fined-grained representations, the network is quite sufficient for accurate location. $z = \varphi\left(z', \theta\right)$ refers to the feature representation, where z' means a search image and θ represents the parameters of the network. Then, representations can be learned via the following objective function:

$$L = \left\| g\left(z'\right) - y \right\|_2^2 + \gamma \|\theta\|_2^2 \tag{5}$$

$$g(z') = Zw = F^{-1}(\hat{z} \odot \hat{w}^*) \tag{6}$$

Here, Z denotes the circulant feature matrix of the search image patch z', F^{-1} represents the Inverse Discrete Fourier transform, and w means the learned CACF based on the CNN features of the target image patch and the global context. The derivatives of the above objective function are then derived.

In order to simplify the derivation process, this paper does not consider the regular term about θ. It can be seen from the objective function that $\frac{\partial L}{\partial x_0}$, $\frac{\partial L}{\partial x_i}$ and $\frac{\partial L}{\partial z}$ must be derived for end-to-end training. Since the intermediate variable in the derivative is a complex number type, the chain rule becomes complicated. Inspired by [32], the partial derivative of Discrete Fourier transform and Inverse Discrete Fourier transform can be written as:

$$\hat{g} = F(g), \frac{\partial L}{\partial \hat{g}^*} = F\left(\frac{\partial L}{\partial g}\right), \frac{\partial L}{\partial g} = F^{-1}\left(\frac{\partial L}{\partial \hat{g}^*}\right) \tag{7}$$

For the back-propagation of the template branch,

$$\frac{\partial L}{\partial x_0} = F^{-1}\left(\frac{\partial L}{\partial \hat{x}_0^*}\right) = F^{-1}\left(\frac{\partial L}{\partial \hat{g}^*}\frac{\partial \hat{g}^*}{\partial \hat{w}}\frac{\partial \hat{w}}{\partial \hat{x}_0^*}\right) \tag{8}$$

$$\frac{\partial L}{\partial x_i} = F^{-1}\left(\frac{\partial L}{\partial \hat{x}_i^*}\right) = F^{-1}\left(\frac{\partial L}{\partial \hat{g}^*}\frac{\partial \hat{g}^*}{\partial \hat{w}}\frac{\partial \hat{w}}{\partial \hat{x}_i^*}\right) \tag{9}$$

For the back-propagation of the search branch,

$$\frac{\partial L}{\partial z} = F^{-1}\left(\frac{\partial L}{\partial \hat{z}^*}\right) = F^{-1}\left(\frac{\partial L}{\partial \hat{g}^*}\frac{\partial \hat{g}^*}{\partial \hat{z}^*}\right) \tag{10}$$

To get $\frac{\partial L}{\partial x_0}$, $\frac{\partial L}{\partial x_i}$ and $\frac{\partial L}{\partial z}$, this paper derives $\frac{\partial L}{\partial \hat{g}^*}$, $\frac{\partial \hat{g}^*}{\partial \hat{w}}$, $\frac{\partial \hat{g}^*}{\partial \hat{z}^*}$, $\frac{\partial \hat{w}}{\partial \hat{x}_0^*}$ and $\frac{\partial \hat{w}}{\partial \hat{x}_i^*}$ as follows.

$$\frac{\partial L}{\partial \hat{g}^*} = F\left(\frac{\partial L}{\partial g}\right) = 2(\hat{g} - \hat{y}) \tag{11}$$

$$\frac{\partial \hat{g}^*}{\partial \hat{w}} = \hat{z}^* \tag{12}$$

$$\frac{\partial \hat{g}^*}{\partial \hat{z}^*} = \hat{w} \tag{13}$$

$$\frac{\partial \hat{w}}{\partial \hat{x}_0^*} = \frac{\hat{y} - \hat{x}_0 \odot \hat{w}}{\hat{x}_0^* \odot \hat{x}_0 + \lambda_1 + \lambda_2 \sum_{i=1}^{k} \hat{x}_i^* \odot \hat{x}_i} \tag{14}$$

$$\frac{\partial \hat{w}}{\partial \hat{x}_i^*} = \frac{-\lambda_2 \hat{x}_i \odot \hat{w}}{\hat{x}_0^* \odot \hat{x}_0 + \lambda_1 + \lambda_2 \sum_{i=1}^{k} \hat{x}_i^* \odot \hat{x}_i} \tag{15}$$

The localization loss is propagated to the convolutional features by the above formulas, and the remaining back-propagation is implemented through the traditional CNN optimization process. The back-propagation of the CACF layer is conducted using Hadamard product in the Fourier domain, so the advantages of the fast computation of the CACF tracker can be maintained, and it can be applied to large-scale datasets for end-to-end training. After completing offline training, CNN features suitable for the CACF tracker are extracted for real-time tracking.

2.3 Channel Attention in DCACFNet

A convolutional feature channel has different effects on tracking different visual objects. It is neither efficient nor effective for object tracking to use all convolutional channels features. Especially, some feature channels play more important roles than the others in certain circumstance. In order to keep the target adaption of deep network under the appearance variation, we incorporate a novel channel attention [31] which is trained using a shallow neural network. The channel attention can be regarded as selecting important convolutional feature channels for different visual objects. The channel attention module is only involved in the temple branch during online tracking, which has little effect on the tracking efficiency.

As shown in Fig. 2, the channel attention consists of global average pooling and two fully-connected layer with a ReLU activation and a Sigmoid activation, respectively. One is used to reduce the channel space of 64 dimensions to 16 dimensions, the other is used to increase the channel space dimension to 64 dimensions. The global average pooling layer encodes global spatial information into a one-dimensional channel descriptor.

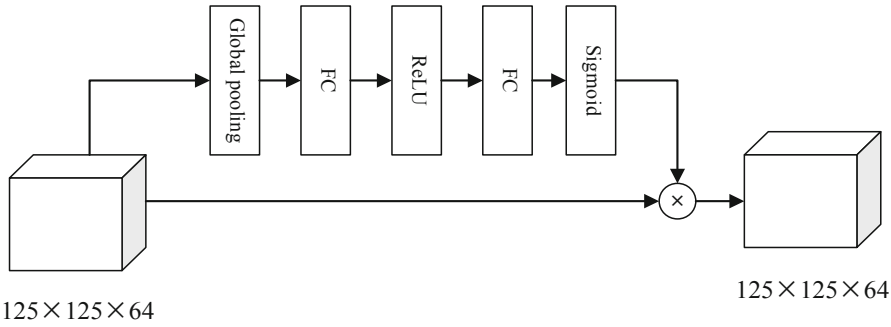


Fig. 2. The Channel Attention architecture

The input of the channel attention module is d channel convolutional features $X = [x_1, x_2, \cdots, x_d]$ with $x_i \in R^{W \times H}$, $W = H = 125, i = 1, 2, \cdots, d$. The output of the channel attention module is denoted by $\tilde{X} = \left[\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_d\right]$ with $\tilde{x}_i \in R^{W \times H}$, $W = H = 125, i = 1, 2, \cdots, d$. This can be achieved by executing the production of a set of channel parameter and the input.

$$\tilde{x}_i = \beta_i \cdot x_i i = 1, 2, \cdots, d \tag{16}$$

Here, β_i represents the parameter of certain convolutional channel.

2.4 High-Confidence Update

Due to the challenging of illumination variation, deformation and occlusion, the appearance of the target often changes during the tracking process. The existing DCF trackers usually update the tracking model every frame without evaluating the tracking result.

Such an update strategy improves the tracking effect to a certain extent, but if the target is located inaccurately or partly occluded, the update strategy is not wise without considering the tracking result, which may cause the total tracking process to fail.

In response to this problem, this paper designs a high-confidence update strategy. The high-confidence update strategy combines the peak value of the response map and the average peak-correlation energy (APCE) [30] to evaluate the tracking results, which is used to determine whether to update the model or not.

The peak value G is the maximum response value in the response map $g\left(z'\right)$, which can be defined as

$$\mathrm{G} = \max g\left(z'\right) \tag{17}$$

We calculated the peak value G of the CACF response map at each frame and grouped them into a set $Q_G = \left\{G^{(2)}, G^{(3)}, \cdots, G^{(t)}\right\}$, which has always no more than 10 elements. The average of Q_G is denoted by $\bar{G}$.

The APCE reflects the fluctuation degree of the response maps, which can be defined as

$$\mathrm{APCE} = \frac{|g_{max} - g_{min}|^2}{mean\left(\sum_{w,h}\left(g_{w,h} - g_{min}\right)^2\right)} \tag{18}$$

Similar to G, we calculated the $APCE$ value of the CACF response map at each frame and grouped them into a set $Q_{APCE} = \left\{APCE^{(2)}, APCE^{(3)}, \cdots, APCE^{(t)}\right\}$, which has always no more than 10 elements. The average of Q_{APCE} is denoted by $\overline{APCE}$.

$$G \geq \alpha_1 \cdot \bar{G} \tag{19}$$

$$APCE \geq \alpha_2 \cdot \overline{APCE} \tag{20}$$

Here, α_1 and α_2 mean the control parameters.

During the tracking process, if the tracking result satisfies the formula (19) and (20) at the current frame, it is high-confidence. Then, the model should be updated with a fixed learning rate β as the formula (21).

$$\hat{w}^t = (1 - \beta)\hat{w}^{t-1} + \beta\hat{w} \tag{21}$$

The peak and the fluctuation degree of the response map can convey the reliability of the tracking results. When the detected target closely matches the correct target, the ideal tracking result should have only one sharp peak and be smooth in all other regions. The sharper the correlation peak, the better the predicting accuracy is. Otherwise, the entire response map will fluctuate sharply, and its pattern will be significantly different from the normal response map.

3 Experiments

This section introduces the experimental details of our DCACFNet at first. Then we perform an experimental analysis on two challenging tracking dataset: OTB-2013[2] with

50 videos and OTB-2015[3] with 100 videos. The experimental results demonstrate the end-to-end trainable discriminative context-aware correlation filter network can improve the tracking performance.

3.1 Implementation Details

Our lightweight network uses VGG-like network as base network, which consists of the convolutional layers, the channel attention module and the CACF layer. Compared to conv1 from VGG, we adopt two convolutional layers ($3 \times 3 \times 64$, $3 \times 3 \times 64$) by removing the pooling layer. The outputs of the temple branch and the search branch are the size of $125 \times 125 \times 64$, which are then fed into the CACF layer to improve localization accuracy. Our whole network is trained on the ILSVRC 2015 dataset [33], which includes more than 4000 sequences and a total of about two million labelled frames. For each sequence, two frames within the nearest 10 frames are randomly picked, which are cropped with double size of the target and then scaled to $125 \times 125 \times 3$. We utilize stochastic gradient descent (SGD) to train the network in an end-to-end way. We train the model for 40 epoch with the learning rate of 10^{-2} and the mini-batch size of 32.

In online track, the hyper-parameters in the CACF layer play a vital role in the tracking performance. The regularization coefficients are set as $\lambda_1 = 0.0001$ and $\lambda_2 = 0.1$. For high-confidence update strategy, the model updating rate is set to 0.01 and two control parameters α_1 and α_2 is set to 0.3 and 0.4, respectively. Meanwhile, the gaussian spatial bandwidth is set to 0.1. Moreover, we adopt 3 scale layers and then set the scale step and the scale penalty to 1.0275 and 0.9925.

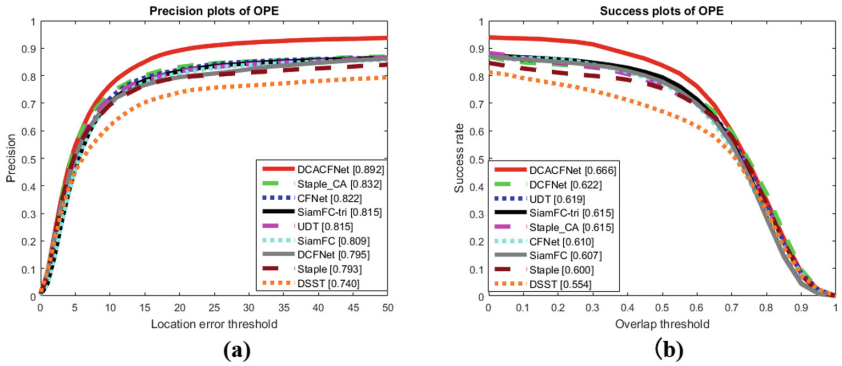
The proposed DCACFNet is implemented with Pytorch framework. All experiments are executed on a PC with a GeForce RTX 1080Ti GPU of 12 GB RAM.

3.2 Experiments Analyses

We evaluate our tracker on the OTB-2013 and OTB-2015 which contain 50 and 100 tracking sequences, respectively. The evaluation metrics consist of precision rate and success rate [2, 3]. The precision rate refers to center location error and the success rate means bounding box overlap ratio.

Comparison on OTB-2013. In OTB-2013 experiment, we test our tracker in comparison with recent state-of-the-art trackers, including UDT (CVPR 2019) [34], SiamFC-tri (CVPR 2018) [35], DCFNet (arXiv 2017) [29], CFNet (CVPR 2017) [28], Staple_CA (CVPR 2017) [22], SiamFC (ECCV 2016) [5], Staple (CVPR 2016) [20], DSST (BMVC 2014) [18]. In Fig. 3 (a) and (b) show the precision and success plot, respectively. Compared with the other eight popular tracking algorithms, DCACFNet in this chapter ranks first in the two indicators of accuracy and success rate of the OTB-2013 dataset, achieving the scores of 89.2% and 66.6%, respectively. It clearly indicates that the proposed tracker, denoted by DCACFNet, achieves competitive performance among these compared trackers in two indications. Compared with the two end-to-end correlation tracking algorithms of DCFNet and CFNet, the proposed DCACFNet outperforms DCFNet and CFNet by 9.7% and 7.0% in the precision rate respectively, because the proposed DCACFNet fully unifies background information, target adaptation, and tracking result.

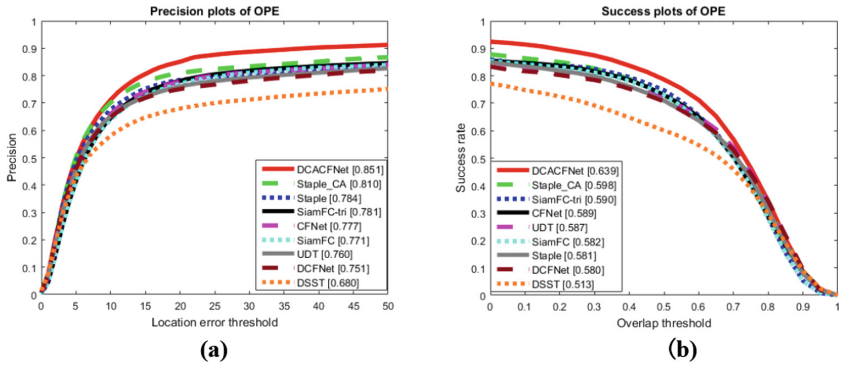
At the same time, the proposed DCACFNet outperforms DCFNet and CFNet by 4.4% and 5.6% in the success rate, respectively.



(a) (b)

Fig. 3. Experimental results on OTB-2013

Comparison on OTB-2015. In OTB-2015 experiment, we compare our tracker against recent state-of-the-art trackers, which include UDT (CVPR 2019) [34], SiamFC-tri (CVPR 2018) [35], DCFNet (arXiv 2017) [29], CFNet (CVPR 2017) [28], Staple_CA (CVPR 2017) [22], SiamFC (ECCV 2016) [5], Staple (CVPR 2016) [20], DSST (BMVC 2014) [18]. In Fig. 4, (a) and (b) show the precision and success plot about these compared trackers, respectively. Compared with the other eight popular tracking algorithms, the proposed DCACFNet ranks first in both the precision and success rates of the OTB-2015 dataset, obtaining the scores of 85.1% and 63.9%, respectively,. It can be seen that our DCACFNet provides the best tracking performance in terms of precision and success metric. In contrast to the integration of correlation filter and Siamese network, our tracker outperforms DCFNet and CFNet by 4.8% and 3.8% in the precision rate respectively because of the fully combination of context-aware correlation filtering, channel attention



(a) (b)

Fig. 4. Experimental results on OTB-2015

mechanism and high confidence update strategy, Meanwhile, our tracker outperforms DCFNet and CFNet by 7.1% and 4.5% in the success rate, respectively.

Attribute-Based Analysis. In order to further analyze detailed performance, we report the results under 11 challenging attributes in OTB-2015, and the results are shown in Fig. 5. The results demonstrate that the tracking performance of our tracker is best under all challenging attributes except for deformation and low resolution. In particular, the success rate under motion blur, fast motion, and out-of-view increase by 8.4%, 9.9%, and 8.3% over the baseline DCFNet, respectively. For the deformation attribute, our tracker obtains a lower success score than the Staple and Staple_CA trackers, which benefit from the complementary of HOG features and color name. Compared to the SiamFC-tri and SiamFC trackers, our tracker achieves a lower success score in the challenging of low resolution due to the usage of the shallow CNN features. Compared with other popular trackers, the overall performance of the proposed DCACFNet is optimal.

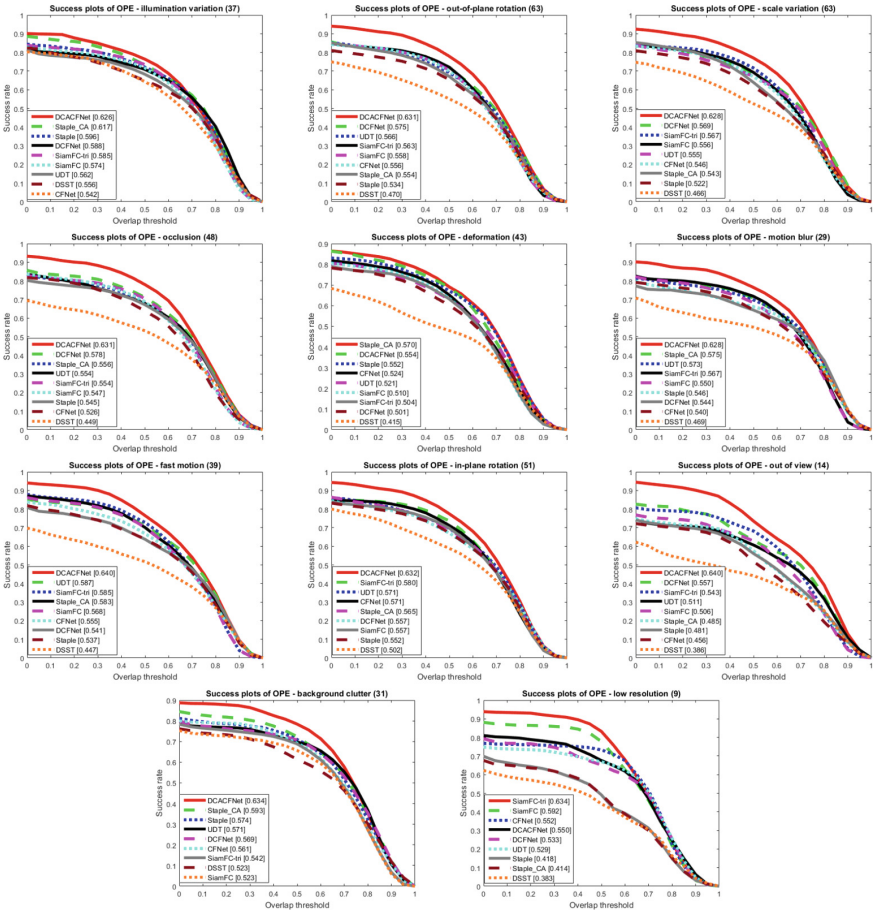


Fig. 5. Success plots obtained by the 9 state-of-the-art trackers about the 11 attributes annotated in OTB-2015.

Qualitative Analysis. For qualitative analysis, Fig. 6 shows the qualitative comparison with recent trackers under four challenging video sequences. As shown in Fig. 6, our DCACFNet is relative successful to track object under some challenging scenarios such as fast motion, motion blur and background clutter, while other trackers hardly cope with these challenging at the same time. For the third row and the fourth row, the proposed DCACFNet successfully locates the target under the challenging of background clutter, while other trackers is poor to track the target.

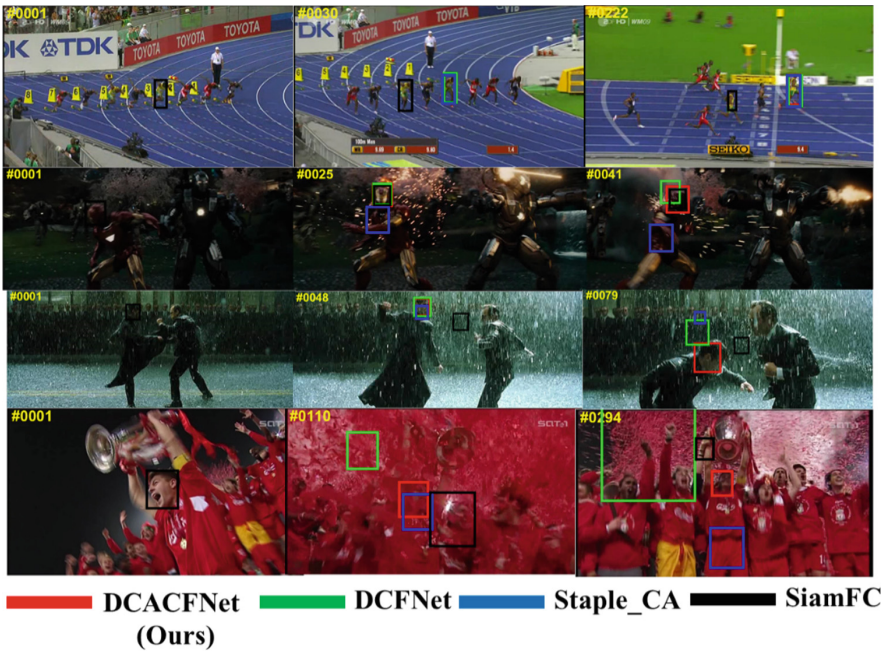


Fig. 6. Qualitative results of our proposed tracker with three state-of-the-art trackers on the bolt, ironman, matrix and soccer video sequences.

4 Conclusion

In this paper, we propose an end-to-end lightweight network architecture for visual tracking which makes use of the global context to improve the tracking performance under the challenging of fast motion and motion blur. The CACF is treated as a special layer which is embedded into the Siamese network. In order to keep target adaption of our tracker, this paper adds a channel attention module after the convolutional layer in the template branch of the DCACFNet framework. A novel high-confidence update strategy is proposed to determine whether to update the model or not, which effectively avoid the model corruption. Evaluations on OTB-2013 and OTB-2015 demonstrate the effectiveness of our approach. In future work, we will pay more attention to the deep level feature representation and the effective feature fusion strategy, which will furthermore improve the robustness of the tracker.

Acknowledgments. This research is partially supported by National Natural Science Foundation of China (No.61876018, No.61976017).

References

1. Kristan, M., Matas, J., Leonardis, A., Felsberg, M.: The visual object tracking VOT2017 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1949–1972 (2015)
2. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: A benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
3. Wu, Y., Lim, J., Yang, M.-H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
4. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
5. Bertinetto, L., Valmadre, J., Henriques, J., Vedaldi, A., Torr, P.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision, pp. 850–865 (2016)
6. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6638–6646 (2017)
7. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
8. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4282–4291 (2019)
9. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: accurate tracking by overlap maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4660–4669 (2019)
10. Ross, D., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comput. Vision **77**(1), 125–141 (2008)
11. Alt, N., Hinterstoisser, S., Navab, N.: Rapid selection of reliable templates for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1355–1362 (2010)
12. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1822–1829 (2012)
13. Avidan, S.: Support vector tracking. IEEE Trans. Pattern Anal. Mach. Intell. **26**(8), 1064–1072 (2004)
14. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007)
15. Bai, Y., Tang, M.: Robust tracking via weakly supervised ranking SVM. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1854–1861 (2012)
16. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2014)
17. Danelljan, M., Khan, F.S., Felsberg, M., Weijer, J.v.d.: Adaptive color attributes for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090–1097 (2014)

18. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, 1–5 September 2014
19. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)
20. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.S.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409 (2016)
21. Bibi, A., Mueller, M., Ghanem, B.: Target response adaptation for correlation filter tracking. In: European Conference on Computer Vision, pp. 419–433 (2016)
22. Mueller, M., Smith, N., Ghanem, B.: Context-aware correlation filter tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1396–1404 (2017)
23. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
24. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 58–66 (2015)
25. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.-H.: Hedged deep tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4303–4311 (2016)
26. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: European Conference on Computer Vision (2016)
27. Bhat, G., Johnander, J., Danelljan, M., Khan, F.S., Felsberg, M.: Unveiling the power of deep tracking. In: European Conference on Computer Vision, pp. 483–498 (2018)
28. Valrnadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2805–2813 (2017)
29. Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W.: Dcfnet: discriminant correlation filters network for visual tracking, arXiv: 1704.04057 (2017)
30. Wang, M.M., Liu, Y., Huang, Z.Y.: Large margin object tracking with circulant feature maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4021–4029 (2017)
31. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
32. Boeddeker, C., Hanebrink, P., Drude, L., Heymann, J., Haeb-Umbach, R.: On the computation of complex-valued gradients with application to statistically optimum beamforming. arXiv: 1701.00392 (2017)
33. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 231–252 (2015)
34. Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., Li, H.: Unsupervised deep tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1308–1317 (2019)
35. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: European Conference on Computer Vision, pp. 472–488 (2018)

Basic Urinal Flow Curves Classification with Proposed Solutions

Dominik Stursa(✉), Petr Dolezel, and Daniel Honc

University of Pardubice, Pardubice, Czech Republic
dominik.stursa@upce.cz
http://www.upce.cz/fei

Abstract. Nowadays, the pressure on prevent invasive methods for diagnostics is still increasing in the health care sector. In the case of the lower urinary tract, early diagnosis can play a significant role to prevent a surgery. Here, the widely used non-invasive test, the uroflowmetry, is observed. As the new measurement devices are being created, new algorithms for basic urinary flow classification must be developed. There, the feature extraction methods are developed and introduced for further use in combination with standard classifiers based on machine learning. In the further work, the methods will be reviewed on extensive dataset, which is currently being created. As the credible dataset verified by several urologist will be obtained, the proposed methods should be examined. Direction of further development will depend on the results of introduced methods.

Keywords: Uroflowmetry · Urine flow classification · Feature extraction · Pattern recognition

1 Introduction

Nowadays, the pressure on prevent invasive methods for diagnostics is still increasing in the health care sector. In the case of the lower urinary tract, early diagnosis can play a significant role to prevent a surgery. Moreover, the percentage of men with lower urinary tract symptoms over age of 40 is slightly rising and counts 60% [8]. As so, the voiding dysfunction should have major impact on the life quality for a large proportion of men. The diagnosis is often based on urinary flow measurements evaluation, called uroflowmetry. The uroflowmetry (UF) is widely used non-invasive test, which evaluates emptying of the bladder [4].

The UF is basically carried out on an outpatient basis, at specific process involving the person urinate into the measurement device (uroflowmeter) at predetermined time. This procedure is not comfortable for the most patients. Fore-more, the voiding "on-demand" is unnatural, which causes the significant test-to-test variability [13]. As so, the test repetition is recommended, which is time-consuming and costly ineffective [6,12]. Therefore, the demand for new small devices for comfortable use, but also for easy and practical methods signaling emerging problems, is growing.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 737–746, 2021.
https://doi.org/10.1007/978-3-030-55180-3_56

The one of mostly used urodynamic measurement method in clinical environment rests on weighing of the voided urine in time [19]. Since the dominant methods for plausible diagnoses are based on the urine flow, the derivative of weighted urine in time must be determined or calculated. Block diagram of the typical uroflowmeter widely used in clinical environment is shown in Fig. 1.

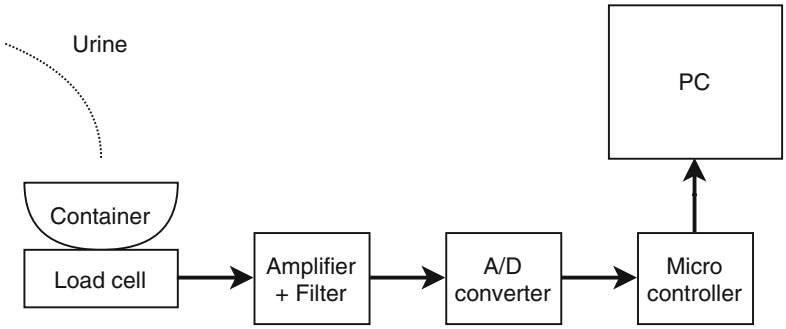


Fig. 1. Block diagram of uroflowmeter, where A/D converter stands for analog to digital converter and PC is personal computer.

When the urinary flow curves are obtained, the significant parameters are measured according to the recommendations of the ICS - International Continence Society [18]. Not only parameters are tracked, but it's also highly recommended to evaluate the whole urinary flow curve. At first, the curves can be divided into continuous and discontinuous types. For more, the continuous curves are divided into normal, box and compressive type [7]. These clusters often indicate a disease. However, since the curve evaluation is mostly performed by urologist or other specialist, the result may be affected by human error. Therefore, the obvious direction of the research is to develop and to propose a method capable of helping the doctor make decisions. Here, the authors try to propose a set of possible machine learning approaches in order to provide the mentioned capability.

In this article, the curves of the urinary flow and the urine weight in time, acquired by mostly used clinical uroflowmeter [19] are observed.

The article is structured as follows. At first the problem is formulated. This includes the description and examples of the typical UF curve shapes and parameters. In the next section, the proposed methods are introduced. Then the possible use is discussed. The article is finished with conclusions.

2 Problem Formulation

The lower urinary tract symptoms are diagnosed with urodynamic methods by default. Urodynamics is a study, where bladder and urethra are on the scope of view when performing their function of storing and releasing urine. One of

mostly used method is an uroflowmetry. The UF is widely used, because of its benefits like non-invasiveness, easy reproduction, simple use, and possibility of repetition over time [3]. The aim of this paper is to propose and discuss a chain of steps necessary for the UF autonomous application.

The UF is widely researched topic based on analysis of bladder emptying divided into a two main branches. The first one is the analysis of the urinal flow curves, where mainly the shapes are observed. The second direction is about analysis of quantitative parameters. Casually, the measured data must be preprocessed for further use in mentioned directions.

2.1 Curves Selection

The uroflowmetric curves measured in typical case cannot be automatically used before preprocessing. It is caused by voiding in unnatural conditions, where the beginning of urination is often repeated, total voiding amount is below minimal limit and the psychical influence occurs when patient urinates in the presence of another person (nurse, doctor). Thus, the only data where total volume of urine is bigger than 150ml are selected for further consideration as recommended in [16]. Then, from the whole measurement process, only the part from real start of urinating and its finish are chosen. As the raw data are considerably affected by noise, the filtering is executed. The curve selection and preprocessing for possible automatic processing is implied in Fig. 2.

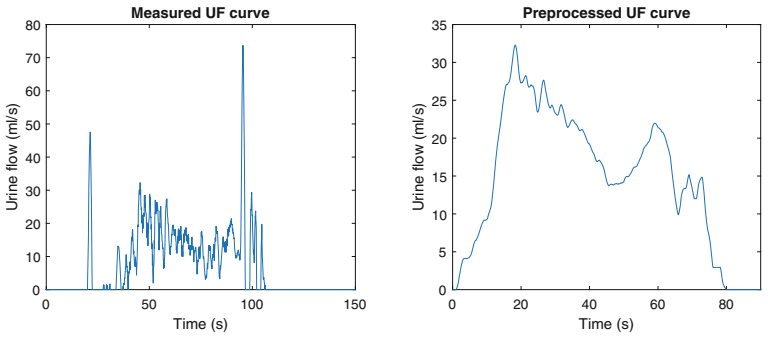


Fig. 2. Measured data from uroflowmeter and preprocessed data for possible use.

2.2 Curve Types

The basic diagnoses are based on the shape of the uroflowmetric curve. The uroflowmetric curve shows the flow of urine through the urethra captured in time. The standard uroflowmeters are measuring the total weight of urine voided into the container in defined time steps. This measurements must be processed to obtain the uroflowmetric curve. The curve shape for health patient is visibly different with patients with any lower urinary tract symptoms (LUTS).

The normal uroflometric curve, representing the patient with healthy lower urinary tract, have a typical bell shape.

Among the frequently observed LUTS, compressive and constrictive type can be mentioned, as was implied in Introduction section. The constrictive type arises with urethra stricture, where the curve have the box shape. The uroflowmetric curve of patient with enlarged prostate (BPH obstruction) has a special shape in between of bell and box shapes. The flow of urine in time graphs for normal and both mentioned obstructions are captured in Fig. 3.

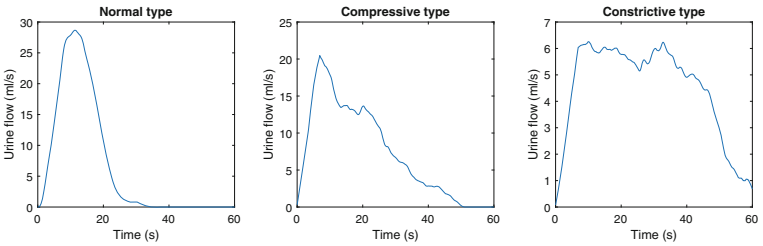


Fig. 3. Uroflowmetric curves.

2.3 Quantitative Parameters

For a simplified information, only several parameters, summarizing main indicators of urine voiding process, can be used instead of the whole curve. The total volume of voided urine is the first parameter. As next, the maximal and average flow can be obtained as suitable parameters. As last major parameters, the urination time and time to reach maximum flow can be selected. These parameters are typically read from uroflowmetric curve by urologist, but could be also numerically calculated from data.

3 Proposed Methods

As the types of curves are visually separated by urologist, the use of classification and clustering algorithms comes in consideration. Not only with the curves, but also with the quantitative parameters or other feature extraction technique, can be possibly the classification realized in practice. As the prevention plays a significant role, the main purpose should be just to separate the bell shape curves (healthy LUT) from others. Accordingly, the pattern recognition techniques should be applied.

3.1 Pattern Recognition System

The process of recognizing patterns with machine learning methods is called pattern recognition. The pattern recognition algorithms are basically composed

of sensing part, preprocessing part, feature extraction algorithm and description algorithm [15].

In pattern recognition system, the preprocessing is a process where data are divided into multiple segments based on obvious data difference. In the case of UF curves, the segmentation is based on LUT symptoms, which are changing curve shape.

The feature extraction mechanism starts on initial set of measured data and builds derived values intended to be informative and non-redundant. The different types of feature extraction applicable to UF data are described in the next section.

3.2 Feature Extraction

As the UF curve is a typical graph containing a dependent variable changing in time, the different approaches of feature extraction could be realized. Fore-more, the urinal system should be described by the VBN model [20]. Here, individual feature extraction approaches are examined. For all possible approaches data must be preprocessed as described in Sect. 2.1 and normalized.

Raw Data as Features. This proposed approach is based on the idea that whole sequence of data could carry enough information for specific pattern recognition [17]. As so, the idea is about putting the normalized UF curve data to description algorithm with its output pairs represented by assigned class. The classes are sufficiently separable, which increases possibility of time series classification by this approach. The scheme is shown in Fig. 4.

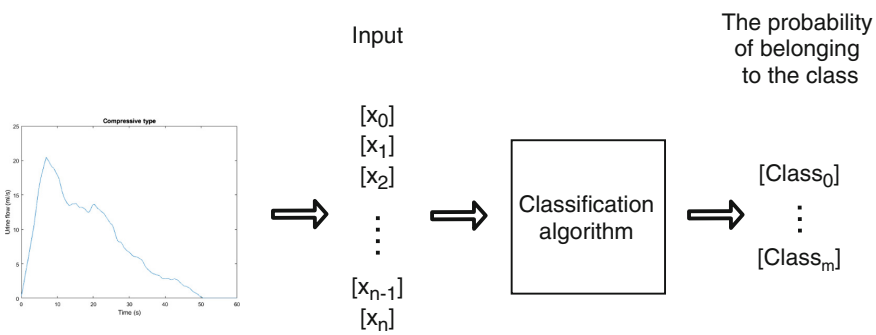


Fig. 4. Time series classification with raw data as features.

Model Parameters Estimation. From the physical essence, the weight of voided urine in time is some kind of transition characteristics. Thus, the general mathematical model of LUT can be defined. Model parameters should vary for each person. However, the correlation of parameters between patients with same symptoms could be detected.

One of methods is to try the data for model identification [2]. For this purpose the weight in time dependency data could be used. At first, the type and the complexity of the model should be defined. ARX, ARMAX, NARX, NARMAX or neural models are only a few of many possible architectures to be selected. The mean square error between data and model can be also tested. Based on minimal error, the best model structure can be selected.

When the model structure is defined, the model parameters could be obtained by model training or fitting. As an input, a step function should be used with max value equal to total weight of voided urine and starting in time of voiding start. Based on model structure, the output could be affected by its previous values. Then, the model parameters should be used as an input to classification algorithm. An example of this approach, considering a neural model, is implied in Fig. 5.

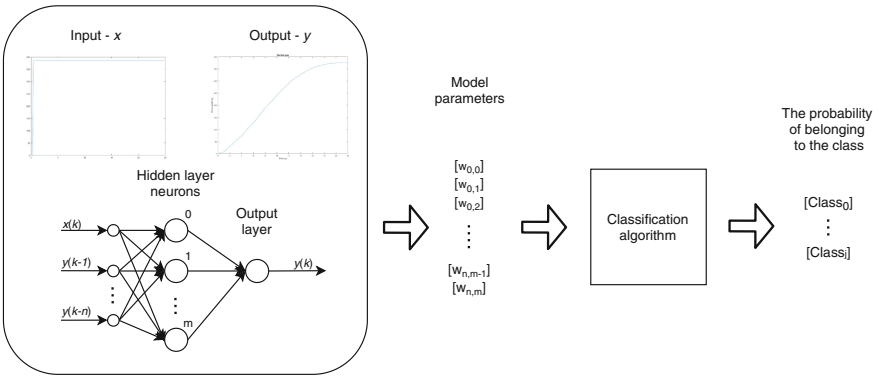


Fig. 5. Schema of the classification using parameters from trained neural model as an inputs.

Polynomial Approximation. Based on same data, the characteristics can be interpolated with defined lines or should be approximated by splines or other mathematical functions. Finding the parameters is an optimization problem often solved by minimization of the mean square error. Subsequent to the data acquisition, the parameters of the splines or mathematical functions can be considered as an input for classification algorithm.

Quantitative Parameters. The quantitative parameters of UF curves are widely used as initial assessment of healthy patient in practice. One of the ideas is to test the possible classification with only quantitative parameters as an input data (Fig. 6).

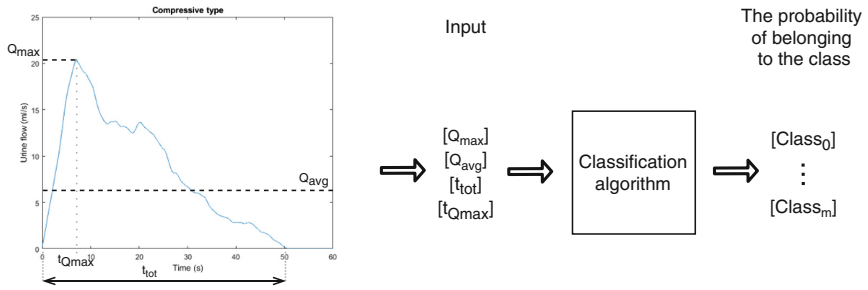


Fig. 6. Quantitative parameters used as features for classification.

3.3 Description Algorithm

Providing a reasonable answer for all possible inputs is generally the main aim of pattern recognition systems. That can be basically achieved using classification methods, clustering or regression methods. As the classes are defined by urologist, the classification methods are on the scope of the article.

Classification Trees. The classification tree methods are used to predict class membership of a categorical dependent variable from their measurements on one or more predictor variables [14]. Predictor or ordered predictor variables categorical splits could be easily realized by binary trees. Not as only at binary decision, but also on linear combination splits the classification can be compute.

Nearest Neighbor. In the pattern recognition, the k-nearest neighbor algorithms are used. It is a non-parametric method used for regression and classification. The input consist of the k closest training examples in the feature space [1]. In the classification problem, the output is a class membership. An object is classified due to voting count of its neighbors, where it is assigned to the most common class among its k nearest neighbors.

Naive Bayes. The naive Bayes classification algorithms are based on the Bayes' theorem, which describes the probability of event, based on prior knowledge of conditions that might by related to the event [11]. Naive Bayes is a simple technique for constructing classifiers, which is the model that assign class labels to problem instances, represented as vectors of feature values.

SVM Classification. The main objective of the support vector machine algorithm is to find a hyperplane in a N-dimensional space, where the N is the number of features, that clearly classifies the data points [5].

Feedforward Neural Networks. The main ability of the neural networks is that they can learn complex non-linear input-output relationships. That is achieved by the sequential training procedures and by great self adeptness to the data [10]. The most commonly used family of the neural networks for the classification task is the feedforward neural network [9].

The feedforward neural network (FFNN) is composed by the input layer, where every input variable must be connected to its neuron, then by the hidden layers and finally by the output layer, where the count of neurons in the output layer is the same as count of outputs. The learning process involves updating of neuron connections weights, which makes the FFNN capable to perform clustering and classification tasks.

4 Conclusion

In the article were presented methods and designs for an automatic uroflow curves recognition and classification. Urologists often use only several indicators on the intuitive bases for the urinary flow curve classification, which in hand with knowing of patient history could lead to clear LUTS diagnoses. According to this idea, the use of only classical recognition methods could finish with failure. As such, the main objective is to use a different approaches for features extraction. The probability of successful LUTS diagnose is increasing with the feature extraction methods diversity. The presented methodology and designs tries to keep the same idea and for that could provide the base for the automatic uroflow curves classification.

5 Discussion and Future Work

As a lot of the clinical measurement devices uses weighting of the voided urine carried out on outpatient basis, the urinary flow curve can be considerably distorted. The data acquisition in natural environment should be preferred, even at the cost of reduced quality caused by unprofessional measurement. Hence, the automatic classification could help with LUT diagnoses based on knowledge of the patient LUT model.

For possible classification of lower urinary track symptoms a several approaches were introduced. In the further work, the methods will be reviewed on extensive dataset, which is currently being created. As the credible dataset verified by several urologist will be obtained, the proposed methods should be examined. Direction of further development will depend on the results of introduced methods. If the results of the methods are comparably accurate with the urologist decisions, their further extensions will be researched. In the other case, the new concepts for LUTS classification must arise.

Acknowledgment. The work has been supported by the IGA Funds of the University of Pardubice, Czech Republic. This support is very gratefully acknowledged.

References

1. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
2. Billings, S.A.: Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains. Wiley, Chichester (2013)
3. Buresova, E., Vidlar, A., Student, V.: Uroflowmetrie, nenahraditelna vysetrovaci metoda k diagnostice mocovych dysfunkci. Urol. Pract. **14**(4), 170–172 (2013)
4. Chua, M.E., et al.: A critical review of recent clinical practice guidelines on the diagnosis and treatment of non-neurogenic male lower urinary tract symptoms. CUAJ-Canad. Urol. Assoc. J. **9**(7–8), E463–E470 (2015)
5. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
6. de la Rosette, J.J.M.C.H., et al.: Relationships between lower urinary tract symptoms and bladder outlet obstruction: results from the ICS-"BPH" study. Neurourol. Urodyn. **17**(2), 99–108 (1998)
7. Drake, M.J., Doumouchtsis, S.K., Hashim, H., Gammie, A.: Fundamentals of urodynamic practice, based on International Continence Society good urodynamic practices recommendations. Neurourol. Urodyn. **37**(6), S50–S60 (2018)
8. Irwin, D.E., et al.: Population-based survey of urinary incontinence, overactive bladder, and other lower urinary tract symptoms in five countries: results of the EPIC study. Eur. Urol. **50**(6), 1306–1315 (2006)
9. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: a tutorial. Computer **29**(3), 31–44 (1996)
10. Jayanta, D.B., Kim, T.-H.: Use of artificial neural network in pattern recognition. Int. J. Softw. Eng. Appl. **4**, 23–33 (2010)
11. Josang, A.: Generalising Bayes' theorem in subjective logic. In: 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) (2016)
12. Kranse, R., van Mastrigt, R.: Causes for variability in repeated pressure-flow measurements. Urology **61**(5), 930–934 (2003)
13. Krhut, J., et al.: Comparison between uroflowmetry and sonouroflowmetry in recording of urinary flow in healthy men. Int. J. Urol. **22**(8), 761–765 (2015)
14. Lahoti, S., Mathew, K., Miner, G.: Tutorial H - predictive process control: QC-data mining using statistica data miner and QC-miner. In: Nisbet, R., Elder, J., Miner, G. (eds.) Handbook of Statistical Analysis and Data Mining Applications, pp. 513–530. Academic Press, Boston (2009)
15. Little, M.A.: Machine Learning for Signal Processing: Data Science, Algorithms, and Computational Statistics. Oxford University Press, New York (2019)
16. Madersbacher, S., Alivizatos, G., Nordling, J., Sanz, C.R., Emberton, M., de la Rosette, J.J.M.C.H.: EAU 2004 guidelines on assessment, therapy and follow-up of men with lower urinary tract symptoms suggestive of benign prostatic obstruction (BPH guidelines). Eur. Urol. **46**(5), 547–554 (2004)

17. Pan, L., Meng, Q., Pan, W., Zhao, Y., Gao, H. A feature segment based time series classification algorithm. In Li, J.B. (ed.) 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), International Conference on Instrumentation Measurement Computer Communication and Control, Qinhuangdao, Peoples R China, September 18–20, pp. 1333–1338. Harbin Inst Tech; Yanshan Univ; NE Univ Qinhuangdao; IEEE Instrumentation and Measurement Soc.; IEEE IM Soc., Beijing & Harbin Joint Chapter; Heilongjiang Instrument and Measurement Soc.; IEEE Computer Soc. (2015)
18. Schafer, W., et al.: Good urodynamic practices: uroflowmetry, filling cystometry, and pressure-flow studies. Neurourol. Urodyn. **21**(3), 261–274 (2002)
19. Urbonavicius, B.G., Kaskonas, P.: Urodynamic measurement techniques: a review. Measurement **90**, 64–73 (2016)
20. Valentini, F.A., Besson, G.R., Nelson, P.P., Zimmern, P.E.: Clinically relevant modeling of urodynamics function: the VBN model. Neurourol. Urodyn. **33**(3), 361–366 (2014)

Mixing Deep Visual and Textual Features for Image Regression

Yuying Wu[1] and Youshan Zhang[2(✉)]

[1] Econometrics, Liaoning University, Shenyang, Liaoning, China
Wuyyyyy@outlook.com
[2] Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA
yoz217@lehigh.edu

Abstract. Deep learning has been widely applied in the regression problem. However, little work addressed both visual and textual features in one unit frame. In this paper, we are the first to consider the deep feature, shallow convolutional neural network (CNN) feature, and textual feature in one unit deep neural network. Specifically, we propose a mixing deep visual and textual features model (MVTs) to combine all three features in one architecture, which enables the model to predict the house price. To train our model, we also collected large scale data from Los Angeles of California state, USA, which contains both visual images and textual attributes of 1000 houses. Extensive experiments show that our model achieves higher performance than state of the art.

Keywords: Deep feature · Textual feature · Convolutional neural network · House price prediction

1 Introduction

In recent years, the real estate market continuously attracts attentions of government and public. The real estate industry gradually becomes an essential pillar of national economy in each country and a necessary part in the development of federal and local economies [12]. Although it has made a great contribution to the growth of economy, it also brings lots of issues, such as, the high-speed rising of housing prices has made many low-income groups in a dilemma since they cannot afford it, and the contradiction between the rich and the poor becomes increasingly prominent. Housing price becomes the focus of various social problems, and it can threaten the sustainable development of society. Moreover, the oscillation of house prices not only affects the value of asset portfolio for most households but also affects the profitability of financial institutions and the surroundings of financial system [1]. The house renovation and construction can also cause difficulty for householder.

The prediction of house price is beneficial for individual investors and they can more intuitively analyze the real estate market. More importantly, it can help the government to regulate the real estate market reasonably and make the

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 747–760, 2021.
https://doi.org/10.1007/978-3-030-55180-3_57

house price more reasonable and the development of the real estate market more stable. Therefore, it is important and necessary to predict the house price with high accuracy.

However, due to the low liquidity and heterogeneity of the real estate market from both physical and geographical perspective, and there are many factors that can affect the house price, they bring great challenges of house price prediction. Although previous studies analyzed the relationship among social, economic, demographic changes and characteristic house prices, quantifying such a correlation is not enough to accurately estimate or predict house prices [17,23]. In addition, most previous work focused on textual features of house price prediction [10,14]. Although Ahmed et al. 2016 first combined the visual and textual features, they have a limited sample size and less textual attributes [1].

In this paper, we address these challenges within one unit deep neural network frame. Specifically, we address the house price prediction issue through considering both deep visual and textual features with exploring several pre-trained deep neural networks.

Our contributions are three-fold:

- We provide a large scale house price prediction dataset, it combines deep visual and ten textual features. The dataset can be of significant value in future research for the community.
- We are the first to propose one unit deep neural network to consider the deep feature, shallow CNN feature, and textual feature. The designed architecture can output the house price with less mean absolute percentage error.
- We are the first to propose a novel absolute mean discrepancy loss to insure that mean predicted price closes to actual mean house price.

Extensive experiments on competitive benchmark housing datasets show that ResNet50 is the best neural network for feature extraction in house price prediction problem.

2 Related Work

House price prediction is a regression problem that aims to output house price given either visual or textual features as input. In the past decade, many methods are proposed for regression problem, such as support vector regression (SVR) [1,16,27], time-series forecasting [6,26], artificial neural network [13,33], deep neural networks [2,31,34], etc.

There are also many studies addressed house price prediction problem. Limsombunchai et al. proposed an artificial neural network (ANN) to predict house prices using 200 house information in Christchurch, New Zealand, and they found that their ANN model significantly improved its predictive power than Hedonic price model [13]. Khamis et al. compared the performance between the multiple linear regression (MLR) model and neural network (NN) model for predicting the house price. They concluded that the prediction of the house price in the NN

model is closer to the ground truth, comparing with the MLR model. Nevertheless, there are some limitations in their model. Firstly, the used house price is not the actual sale price but the estimated price. Secondly, this study considered only the specific year's information on the houses and ignored the time effect of the house price. Finally, the house price could be affected by some other economic factors that are not included in the estimation [10]. Ahmed et al. collected 535 sample houses in California, USA, which contained both visual images and textual features. They compared the NN and support vector machine methods on the effect of predicting house prices, and they observed that using NN can achieve better results than SVM model. In addition, they found that compared to the individual textual features, the combination of both visual and textual features produced better estimation accuracy. However, they had less training data, which may lead to overfitting; they also used a lower level feature, which is from SURF feature extractor, and deeper neural networks can be applied to extract features [1]. Nguyen identified the four most essential attributes in housing price prediction across three counties that are assessment, comparable houses' sold price, listed price, and the number of bathrooms [15]. Varma et al. considered another seven features that affect house price (area, number of bedrooms, number of bathrooms, type of flooring, lift availability, parking availability, and furnishing condition). Differing from the traditional hedonic house price model, they focused on spatial econometrics and cross-validates the out-of-sample prediction performance of 14 competitive models, which contributes to house price prediction. Their results indicated that a nonlinear model, which considered the spatial heterogeneity and flexible nonlinear relationship between a specific individual or regional characteristics of a house and its price is the best strategy for predicting house prices [24]. Wang et al. designed a multilayer feed-forward neural network with a memory resistor, which realized automatic online training. Memristor weights of the ANN can be adjusted by the BP algorithm to build up a regression model simultaneously. The neural network is trained and predicted by using the housing price samples of several cities in Boston, USA, and the predicted results close to the target data [25]. Gao et al. noted that the location of the buildings is the most critical attribute that affects the house price, and they defined and captured a fine-grained location profile powered by a diverse range of location data sources [5].

However, none of these work addressed both visual and textual features in one unit deep neural network. First of all, for the visual feature in image regression, the community focused on feature representation from the hand-crafted features to the extracted features from the deep pre-trained neural networks. Before the rise of the convolution neural network, hand-crafted features were well used (e.g., SURF [1]), and since deep features can substantially improve the performance of domain adaption, they are now widely used in both classification [30,35] and regression problem [4,31,34]. Secondly, our work include more influential factors of house price. Furthermore, we investigate both deep visual and textual features on house price estimation in one unit frame.

The rest of this paper is organized as follows. Section 3 describes our collected house data. Section 4 first defines the house price prediction problem and then describes the proposed network architecture and objective function for solving the optimization problem. Section 5 reports the experimental results. We further discuss the results in Sect. 6 and conclude in Sect. 7.

3 Los Angeles Housing Dataset

We collected house images and their associated attributes from www.zillow.com. The data consists of 1,000 sample houses from Los Angeles, California state in the United States. We collected both visual and textual information from houses. For visual features, we saved most representative images of houses, that each house has four different views: frontal, bedroom, bathroom, and kitchen. Figure 1 shows one house image from four different views.

For textual features, there are nine factors in our collected data (number of bedrooms, number of bathrooms, area, zip code, year build, year renovation, house type, parking space, and sun number). Therefore, we have a total of ten attributes on one house, including the house price. Table 1 lists the detailed information of the collected dataset (mode is reported for Zipcode attribute).

We also extract deep visual features from pre-trained models. Figure 2 shows the extracted feature vectors using pre-trained ResNet50 model. It also presents both visual and textual features of the house.

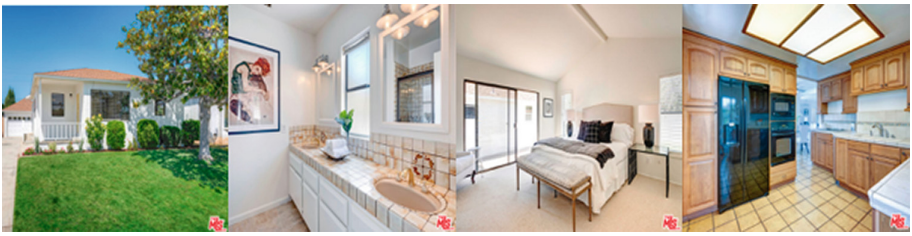
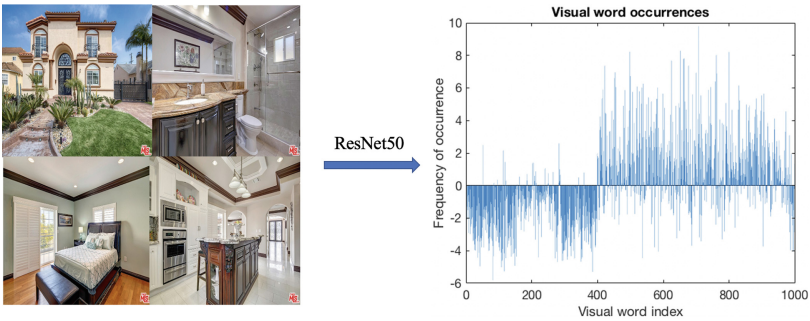


Fig. 1. One sample house image from (www.zillow.com), it is represented by four different views: frontal, bathroom, bedroom, and kitchen.

4 Methods

4.1 Problem and Notation

For house price prediction, it is a regression problem. Given house data (including visual images and textual factors) $\mathcal{X}$ with its associated house price $\mathcal{Y}$; our ultimate goal is to predict the house price ($\mathcal{Y}'$) by given $\mathcal{X}$, which closes to the actual house price $\mathcal{Y}$, that is to minimize the difference between the $\mathcal{Y}'$ and $\mathcal{Y}$.



No. Bedroom	No. Bathroom	Area	Zip code	Year build	Year renovation	Type	Parking spaces	Sun Number	Price
5	3	3374	90045	2007	2015	0	3	90.94	1.959M

Fig. 2. Sample features from one house. The top left is the original image, and top right is the extracted feature from the ResNet50 model, which shows the frequency of occurrence of features. In the bottom, we list the ten attributes of house (Area: square ft, and there are five types of the house (0: single family, 1: townhouse, 2: condo, 3: multi-family, 4: apartment), M: million).

Table 1. Some detailed statistics of Los Angeles housing dataset

Details	Minimum	Maximum	Average
No. bedrooms	1	32	3.52
No. bathrooms	1	12	2.9455
Area (sqft)	528	31520	2211.6
Zipcode	90020	90292	90045 (mode)
Year build	1902	2019	1952.4
Parking space	0	8	2.162
Sun number	22.56	95.75	86.5714
House price ($)	0.2295 M	10.9 M	1.6813 M

4.2 Mixing Deep Visual and Textual Features Model (MVTs)

The architecture of our proposed MVTs model is shown in Fig. 3, we have three modules in the model. First of all, we extract deep feature from a pre-trained ResNet50 model. Second, we design the shallow CNN module to extract the shallow feature from raw images. Third, we add textual feature, that includes the number of bedrooms, the number of bathrooms, area, zipcode, year build, year renovation, house type, parking space, and sun number. We then concatenate all three modules together to form our MVTs model[1].

Our model minimizes the following objective function:

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}) = \arg\min \quad \mathcal{L}_{\mathcal{M}}(\mathcal{X}, \mathcal{Y}) + \alpha \, \mathcal{L}_{\mathcal{A}}(\mathcal{X}, \mathcal{Y}) \tag{1}$$

[1] Dataset is available at https://github.com/heaventian93/House_Price.

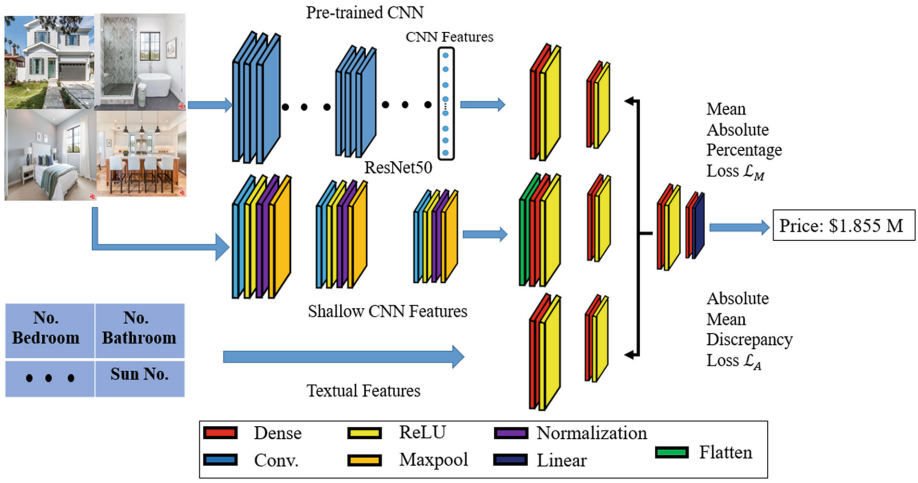


Fig. 3. The architecture of our proposed mixing deep visual and textual features for house price prediction. There are three modules in the model: deep feature module, shallow CNN feature module and textual feature module. In deep feature module, the feature is obtained from the pre-trained ResNet50 neural network. The shallow CNN feature module directly gets features from raw images via three repeated blocks. The textual feature module majorly contains two layers. The model consists of two different loss functions, including mean absolute percentage error loss and absolute mean discrepancy loss. The mean absolute percentage error loss measures mean percentage difference between predicted price and actual house price, and absolute mean discrepancy loss ensures that the mean predicted price approximates the mean house price.

where $\mathcal{L}_{\mathcal{M}}$ is the mean absolute percentage error loss, $\mathcal{L}_{\mathcal{A}}$ is the proposed absolute mean discrepancy loss and α is the balance factor between two loss functions. Specifically ($\bar{\cdot}$ is the mean house price),

$$\mathcal{L}_{\mathcal{M}} = \frac{1}{N}\sum_{i=1}^{N}|\frac{y_i - y_i'}{y_i}|, \quad \mathcal{L}_{\mathcal{A}} = |\frac{\overline{y} - \overline{y'}}{\overline{y}}|. \tag{2}$$

In deep feature module, we followed the protocol from [29,30,32,35], that all features are extracted from the last fully connected layer. Thus the final output of one image becomes one vector 1×1000. Feature extraction is implemented via two steps: (1) rescale the image into different input sizes of pre-trained neural networks; (2) extract the feature from the last fully connected layer. After feature extraction, it follows by two repeated blocks. In each block, it has a dense layer and a "ReLu" activation layer. The units of the dense layer are 1000 and 4, respectively.

In the shallow CNN feature module, it first includes three repeated blocks, and each block has four layers: convolutional 2d layer, "ReLu" activation layer, batch normalization layer and maxpooling layer. In the convolutional 2d layer,

the kernel size is (3, 3) in all three blocks and the filter size is 16, 32, 64, respectively. The pool size in maxpooling layer is (2, 2). After a flatten layer, it then followed by the same block as deep feature module, but the units of the dense layer are 16 and 4.

In the last textual feature module, the input includes nine factors of the house price (number of bedrooms, number of bathrooms, area, zipcode, year build, year renovation, house type, parking space, and sun number). It also followed by the same block as deep feature module, but the units of the dense layer are 8 and 4 since it has fewer features than the other two modules.

We then concatenate all three modules together, it majorly consists of four layers: a dense layer with units number of 4, the "ReLu" activation layer, a dense layer with units number of 1 and finally ends with a "linear" activation layer. Therefore, it can output the house price in the last layer.

There are three obvious advantages of the proposed mixing deep visual and textual features (MVTs) model. First of all, we consider the mixing features from both deep visual and textual features. By contacting all three features, our MVTs model can take advantage of all these features and achieve high performance. In addition, we propose a novel loss function: absolute mean discrepancy loss. It effectively measures the mean difference between the predicted house price and real house price. The performance of this loss function is shown in Sect. 6. Furthermore, the designed Dense and Activation block can successfully leverage deep visual and textual features to predict the house price without overfitting problems. If there are more layers added (e.g., repeated Dense, ReLU, Normalization, and Dropout several times), the final error will be first stuck in a global minimum number and then increased.

5 Results

5.1 Experimental Setting

Our experiment is based on Keras and runs on top of TensorFlow. In addition, our network is trained on a graphics processor NVIDIA Geforce 1080 Ti equipped with 11 Gb of memory on a 16 GB RAM Alienware computer to exploit its computational speed. The network parameters are set to:

1. Batch size: 32
2. α: 10
3. Iterations: 60

We also randomly split our data into training and testing data, and there are 500 samples in both training and testing stages.

5.2 Performance Evaluation

Figure 4 compares the predicted house prices with the actual house prices. We find that the predicted house prices close to the actual house prices, which represents our proposed MVTs is suitable for house price prediction.

Fig. 4. Two examples of house price prediction. The actual price and predicted price are labeled in house images (M: million).

To show the effeteness of our proposed MVTs models on house price prediction, we report both the mean absolute percentage error (MAPE) and the mean square error (MSE) of the testing data (500 samples). MAPE is defined in Eq. 2, and MSE is measured by Eq. 3. Notice that, the smaller of these two metrics, the better the performance of the model.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i), \tag{3}$$

where y_i is the actual house price, and $\hat{y}_i$ is predicted house price.

As shown in Table 2, it lists the MAPE and MSE of different settings of our model. We first run each feature individually to predict the house price in testing data and then report the result of combination of all three features. Across all three different features, the textual feature achieves a higher performance than the other two features. Although textual features have fewer features than deep features, it contains more useful information than just simply the exterior and interior of a house such as the location and area. Moreover, deep feature has a better result than shallow CNN feature since the deep neural network is trained with millions of images. Therefore, we can extract a better feature than the shallow CNN feature. However, a combination of all three features achieves the highest performance (lowest MAPE and MSE). We also observe that all four different variant results are better than the previous SVR model [1].

Furthermore, MAPE is a better metric than MSE. For shallow CNN feature and deep feature, the normalized MSE closes to each other, but there is a significant difference between the two MAPE scores. To further validate this observation, we show the effectiveness of two metrics in Table 3. we find that the MSE is significantly changed if the data is normalized. However, MAPE still

keeps similar as before. As shown in Table 3, it lists the number of iterations and time of different data processing. It suggests that normalized the house price leads to less time, smaller number of iterations and higher performance than the unnormalized data. Therefore, our MVTs model will benefit from the normalized house price.

Table 2. House price results prediction

	Deep feature	Textual feature	Shallow feature	MVTs	SVR
MAPE (%)	42.06	20.93	60.77	**18.01**	72.97
MSE	0.00998	0.00546	0.08012	**0.00477**	0.1304

Table 3. Iterations and time of different data processing

	No. Iterations	times (s)	MAPE	MSE
Unnormalized price	300	2405	34.45	8.87908e11
Normalized price	**60**	**535**	**18.45**	**0.003438**

5.3 Our Model on Ahmed et al. 2016 Housing Dataset

To show the applicability of our model, we also test our model using previous dataset. The housing data is from [1], it consists of 535 house images and we use 75% as training and 25% as the testing. Our model is significantly better than the SVR model as shown in Table 4.

Table 4. House price prediction results of [1]

	MVTs	SVR
MAPE (%)	**34.29**	56.09
MSE (norm.)	**0.01734**	0.1494

6 Discussion

From above results, our proposed MVTs model is able to predict the house price with lower MAPE and MSE values, which demonstrate the robustness of our model. Also, we compare the effectiveness of two different loss functions in Table 5, we find that the performance of single $\mathcal{L}_{\mathcal{A}}$ loss is worse than single $\mathcal{L}_{\mathcal{M}}$ loss since $\mathcal{L}_{\mathcal{A}}$ only measures that mean average of absolute percentage error. However, the combination of these two loss functions achieves the highest performance.

Table 5. House price results with different loss functions

	$\mathcal{L}_{\mathcal{M}}$	$\mathcal{L}_{\mathcal{A}}$	Combination
MAPE (%)	22.10	25.77	**18.01**
MSE	0.00447	0.00589	**0.00477**

Table 6. MAPE and MSE of predicted house price with minimizing $\mathcal{L}_{\mathcal{M}}$ and $\mathcal{L}_{\mathcal{A}}$ using sixteen pre-trained neural networks

Task	MAPE (%)	MSE
SqueezeNet	42.09	0.01005
AlexNet	49.66	0.01095
GoogleNet	51.33	0.00985
ShuffleNet	47.76	0.00943
ResNet18	41.10	0.00884
Vgg16	46.21	0.00869
Vgg19	49.69	0.00942
MobileNetv2	48.18	0.00946
NasnetMobile	48.32	0.00925
ResNet50	**40.58**	**0.00652**
ResNet101	43.92	0.00794
DenseNet201	43.89	0.00830
Inceptionv3	45.76	0.00818
Xception	47.13	0.00882
InceptionresNetv2	49.17	0.00851
NasnetLarge	44.14	0.00869

In addition, we explore how different pre-trained models affected the final results. We examine sixteen well trained models. Specifically, these sixteen neural networks are Squeezenet [9], Alexnet [11], Googlenet [21], Shufflenet [28], Resnet18 [7], Vgg16 [19], Vgg19 [19], Mobilenetv2 [18], Nasnetmobile [36], Resnet50 [7], Resnet101 [7], Densenet201 [8], Inceptionv3 [22], Xception [3], Inceptionresnetv2 [20], Nasnetlarge [36]. As shown in Table 6 and Fig. 5, the ResNet50 surprisingly achieves higher performance than the other models. We further explore the relationship between the top-1 accuracy of sixteen pre-trained neural networks and the MAPE value in Fig. 6. However, we find that the correlation score and R^2 value are relatively smaller. The smaller of these values, the correlation is less [31]. Therefore, we cannot observe a significant trend that how to choose the best pre-trained model via top-1 accuracy. Although we do not know the underlying mechanisms, ReseNet50 is useful in the house price prediction problem.

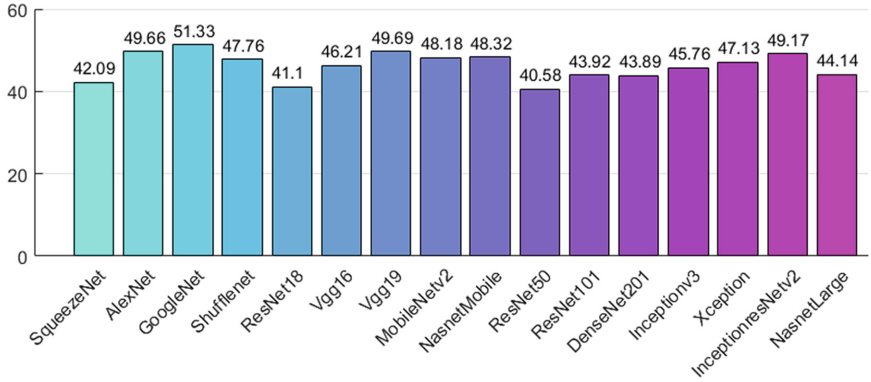


Fig. 5. Bar plot of MAPE values with extracted features from sixteen pre-trained models (x-axis is the order of Top-1 accuracy on ImageNet model).

Table 2 shows that the performance of image features (both deep feature and shallow CNN feature) is significantly lower than the single textual feature. What causes this problem? Actually, the price of house can be affected by lots of factors such as the factors in the textual features. Especially, location is one of the most important factors that affects house prices. However, house images come from four different views, and it only considers the exterior and interior of a house. Therefore, we expect the performance of single deep feature and shallow CNN feature is worse than the single textual feature.

We clearly observe several advantages of our proposed mixing deep visual and textual features model. First of all, our model considers three different features: deep feature from the pre-trained model, shallow CNN feature, and textual feature. Secondly, we propose a novel loss function called absolute mean discrepancy loss, and it can jointly reduce the difference between predicted house price and actual house price.

One interesting phenomenon is that we find the performance of our new data is better than the results of the dataset. One reason is that we collected more images (1000 samples) with 500 samples in the training stage. The second reason is that we collected more factors which can affect the house price. We have nine factors in new house data, while there are only four factors in Ahmed et al. 2016 dataset.

However, our model also has some limitations. Firstly, the lowest MAPE score of testing data is higher than 18%, which implied that the predicted house price can be 18% away from the real price. Therefore, our model can still be improved. Secondly, the MAPE scores of both visual features (deep feature and shallow CNN feature) are relatively lower. This further indicates the deep architecture of images needs to be further investigated.

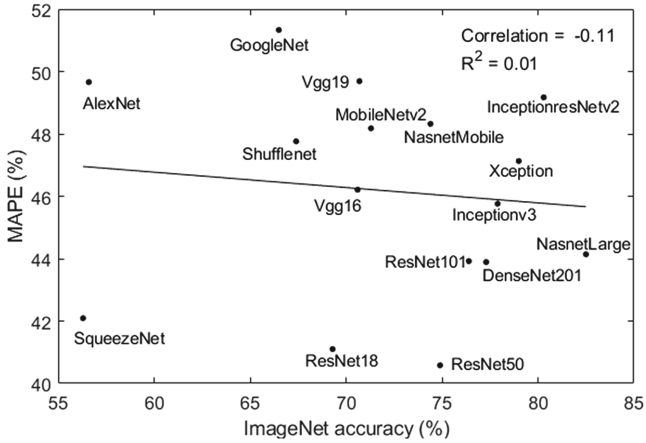


Fig. 6. Correlation between ImageNet accuracy of sixteen pre-trained models and MAPE in the house price prediction.

7 Conclusion

In this paper, we are the first to consider both visual and textual features for house price prediction in one unit architecture. The proposed mixing deep visual and textual features model shows its robustness in house price prediction. Our experiments show that aggregating both visual and textual attributes yielded better prediction results than the deep feature, shallow CNN feature, and textual feature alone. In addition, ResNet50 is the best pre-train model for feature extraction in our dataset.

References

1. Ahmed, E., Moustafa, M.: House price estimation from visual and textual features. arXiv preprint: arXiv:1609.08399 (2016)
2. Akita, R., Yoshihara, A., Matsubara, T., Uehara, K.: Deep learning for stock prediction using numerical and textual information. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6. IEEE (2016)
3. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)
4. Ding, H., Tian, Y., Peng, C., Zhang, Y., Xiang, S.: Inference attacks on genomic privacy with an improved HMM and an RCNN model for unrelated individuals. Inf. Sci. **512**, 207–218 (2020)
5. Gao, G., Bao, Z., Cao, J., Kai Qin, A., Sellis, T., Wu, Z., et al.: Location-centered house price prediction: A multi-task learning approach. arXiv preprint arXiv:1901.01774 (2019)

6. Gupta, R., Miller, S.M.: The time-series properties of house prices: a case study of the Southern California market. J. Real Estate Financ. Econ. **44**(3), 339–361 (2012)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
8. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
9. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
10. Khamis, A.B., Kamarudin, N.K.K.B.: Comparative study on estimate house price using statistical and neural network model. Int. J. Sci. Technol. Res. **3**(12), 126–131 (2014)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
12. Li, Y., Leatham, D.J.: Forecasting housing prices: dynamic factor model versus LBVAR model. Technical report (2010)
13. Limsombunchai, V.: House price prediction: hedonic price model vs. artificial neural network. In: New Zealand Agricultural and Resource Economics Society Conference, pp. 25–26 (2004)
14. Ng, A., Deisenroth, M.: Machine Learning for a London Housing Price Prediction Mobile Application. Imperial College, London (2015)
15. Nguyen, A.: Housing price prediction (2018)
16. Plakandaras, V., Gupta, R., Gogas, P., Papadimitriou, T.: Forecasting the us real house price index. Econ. Model. **45**, 259–267 (2015)
17. Quigley, J.M.: Real estate prices and economic cycles (2002)
18. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
21. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
22. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
23. Tsatsaronis, K., Zhu, H.: What drives housing price dynamics: cross-country evidence. BIS Q. Rev. (2004)
24. Varma, A., Sarma, A., Doshi, S., Nair, R.: House price prediction using machine learning and neural networks. In: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1936–1939. IEEE (2018)
25. Wang, J.J., et al.: Predicting house price with a memristor-based artificial neural network. IEEE Access **6**, 16523–16528 (2018)

26. Wilson, I.D., Paris, S.D., Andrew Ware, J., Harrison Jenkins, D.: Residential property price time series forecasting with neural networks. In: Applications and Innovations in Intelligent Systems IX, pp. 17–28. Springer (2002)
27. Wu, C.-H., Li, C.-H., Fang, I.-C., Hsu, C.-C., Lin, W.-T., Wu, C.-H.: Hybrid genetic-based support vector regression with Feng Shui theory for appraising real estate price. In: 2009 First Asian Conference on Intelligent Information and Database Systems, pp. 295–300. IEEE (2009)
28. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
29. Zhang, Y., Allem, J.-P., Unger, J.B., Cruz, T.B.: Automated identification of hookahs (waterpipes) on Instagram: an application in feature extraction using convolutional neural network and support vector machine classification. J. Med. Internet Res. **20**(11), e10513 (2018)
30. Zhang, Y., Davison, B.D.: Modified distribution alignment for domain adaptation with pre-trainedinception resnet. arXiv preprint arXiv:1904.02322 (2019)
31. Zhang, Y., Davison, B.D.: Shapenet: age-focused landmark shape prediction with regressive CNN. In: 2019 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1–6. IEEE (2019)
32. Zhang, Y., Davison, B.D.: Impact of imagenet model selection on domain adaptation. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops, pp. 173–182 (2020)
33. Zhang, Y., Guo, L., Li, Q., Li, J.: Electricity consumption forecasting method based on MPSO-BP neural network model. In: 2016 4th International Conference on Electrical & Electronics Engineering and Computer Science (ICEEECS 2016) (2016)
34. Zhang, Y., Li, Q.: A regressive convolution neural network and support vector regression model for electricity consumption forecasting. In: Future of Information and Communication Conference, pp. 33–45. Springer (2019)
35. Zhang, Y., Xie, S., Davison, B.D.: Transductive learning via improved geodesic sampling. In: Proceedings of the 30th British Machine Vision Conference (2019)
36. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8697–8710 (2018)

History-Based Anomaly Detector: An Adversarial Approach to Anomaly Detection

Pierrick Chatillon[1(✉)] and Coloma Ballester[2(✉)]

[1] École normale supérieure Paris-Saclay, Cachan, France
pierrick.chatillon@ens-paris-saclay.fr
[2] Universitat Pompeu Fabra, Barcelona, Spain
coloma.ballester@upf.edu

Abstract. Anomaly detection is a difficult problem in many areas and has recently been subject to a lot of attention. Classifying unseen data as anomalous is a challenging matter. Latest proposed methods rely on Generative Adversarial Networks (GANs) to estimate the normal data distribution, and produce an anomaly score prediction for any given data. In this article, we propose a simple yet new adversarial method to tackle this problem, denoted as History-based anomaly detector (HistoryAD). It consists of a self-supervised model, trained to recognize 'normal' samples by comparing them to samples based on the training history of a previously trained GAN. Quantitative and qualitative results are presented evaluating its performance. We also present a comparison to several state-of-the-art methods for anomaly detection showing that our proposal achieves top-tier results on several datasets.

Keywords: Anomaly detection · Generative Adversarial Networks · Wasserstein and total variation distances

1 Introduction

Anomaly detection usually refers to the identification of unusual patterns that do not conform to expected behaviour of data, be it visual data such as images and videos, or other modalities such as acoustics or natural language. Its applications are numerous and include the detection of anomalies in medical or biological imaging such as failure of neurocognitive functions in damaged brains [1–3], real-life image forgery resulting in fake news or even fraud [4–7], anomaly detection in image or video for autonomous navigation, driver assistance systems or surveillance systems for, *e.g.*, violence alerting or evidence investigation [8–13], or for detection of violation and foul in sports analysis, detection of defective samples in manufacturing industry [14–16], sea mines in side-scan sonar images [17] or extrange aerial objects in aerial images that may produce collisions [18], anomalies from multi-modal data including visual data, audio data or natural language [19], to name but a few of its applications.

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 761–776, 2021.
https://doi.org/10.1007/978-3-030-55180-3_58

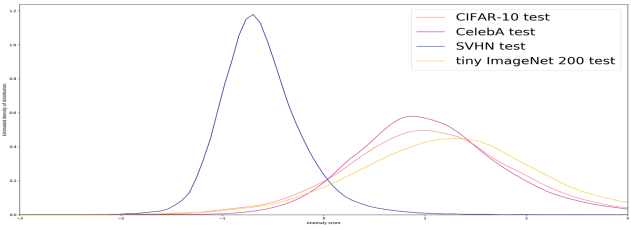


Fig. 1. Method trained on SVHN and evaluated on several datasets.

Table 1. AUPRC for SVHN compared to other datasets.

Test split	CIFAR-10	CelebA	Tiny ImageNet
AUPRC	0.941	0.976	0.949

The precise definition of anomalous data is inherently difficult as, in practice, an unexpected anomaly can be detected only against the ground of a pattern regularity. This is one of the reasons that anomaly detection is frequently approached as out-of-distribution or outlier detection. A detailed account of the many existing methods to approach this problem can be found in [20–23].

This paper proposes a new method for anomaly detection in the context of image processing that is based on the unsupervised learning of the underlying probability distribution of normal data through appropriate GANs and the proposal of a new anomaly score for the detection of abnormal images. Our anomaly detector leverages a recorded history of the normal data generator to fully discriminate regions where true data points are more dense and use this learning to successfully detect anomalies. It results in a general anomaly detector that is free of assumptions on the data and thus it can be applied in any context and data modality. Figure 1 illustrates an example of the performance of our anomaly detector on structurally different datasets. In this experiment, the distribution of the Street View House Numbers (SVHN) dataset [24] is first learned (details in Sect. 3) and considered as normal data. Then, our anomaly score is computed on samples of it and also on samples from the CIFAR-10 [25], CelebA [26] and ImageNet [27] datasets. The approximated density of the anomaly score distribution for each dataset is shown at the top of Fig. 1. Let us notice that, for the normal data, its anomaly values are around -1 while for all the 'anomalous' datasets, the anomaly scores are around $+1$. On the other hand, Table 1 in Fig. 1 bottom shows the area under the precision-recall curve (AUPRC).

The outline of this paper is as follows. Section 2 reviews related research. Section 3 details our proposal. In Sect. 4, the model architecture and implementation details are provided while the experimental results are presented in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related Work

The automatic identification of abnormal or manipulated data is crucial in many contexts [2,4,6,7,10,16,28]. Anomaly detection, has been a topic in statistics for centuries (see [20–23] and references therein). The authors of [22] classify the methods in the literature by the structural assumption made on the normality. Other works challenge anomaly detection with unsupervised or self-supervised learning strategies by taking advantage of the huge amount of data frequently at our disposal [29,30]. Some of them use generative models to learn the (normal) data distribution. Generative models are methods that produce novel samples from high-dimensional data distributions, such as images or videos. Currently the most prominent approaches include autoregressive models [31], variational autoencoders (VAE) [32], and generative adversarial networks (GANs) [33]. GANs are often credited for producing less burry outputs when used for image generation. In the anomaly detection context, several approaches tackle it using autoencoders [34] or GANs [30,35–41] (we refer to [42] for a summary of those GAN-based anomaly detection methods). Some works focus on the implicit inversion of the generator in order to detect anomalous data that do not fall in the learned model [38,43], while others directly infer likelihoods with, for instance, normalizing flows [44–48]. On the other hand, the recent paper [34] uses a memory-augmented autoencoder which learns and records a fixed number of prototypical normal encoded vectors. Given an input sample, it is encoded and the memory is then accessed with an attention-based module to express this encoding by a sparse combination of the stored normal prototypes that used to reconstruct the input data via a decoder. The l_2 distance between the input and its reconstruction is used as anomaly score. Very differently, our anomaly detector leverages the recorded history of the normal data generator to fully discriminate regions where true data points are more dense and use this learning to successfully detect anomalies. The idea of producing an anomaly score prediction for any given data has also been investigated [30,38,39]. Our proposal fits in the class of self-supervised approaches and it is trained only on normal (non-anomalous) samples. The proposed method is general, efficient and simple as it uses the rich information of the training process in the construction the anomaly detector.

3 Proposed Method

We will attribute an anomaly score to any image. Inspired by some ideas in [40] and [41], this score consists of the output of a network.

More precisely, let $\mathbb{P}_{\text{data}}$ be the probability distribution of a given 'normal images' dataset. Our proposal is grounded on, first, the learning of the probability distribution $\mathbb{P}_{\text{data}}$ using a GAN learning strategy while simultaneously keeping track of the states of the associated generator and discriminator during training. Secondly, we create a probability distribution (denoted as $\mathbb{P}_{G_{\text{hist}}}$) that combines different states of the previous generator's history. We finally train

our anomaly detector by computing the Total Variation distance between the real data distribution $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$. Figure 2 displays an outline of the whole method, and in the following Sects. 3.1, 3.2, and 3.3, these steps of our approach are detailed and justified.

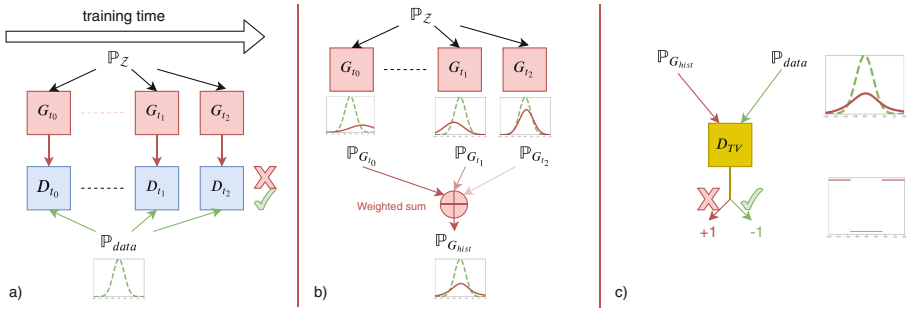


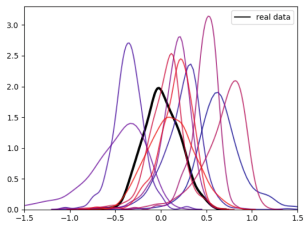
Fig. 2. Outline of the proposed method: a) Some states of the generator are saved during GAN training (G_{t_i} and D_{t_i} represent the states of the generator and discriminator at training time t_i). b) these networks are used to form a new distribution $\mathbb{P}_{G_{\text{hist}}}$ rich in 'anomalous' samples. c) We use this distribution as negative class for a classifier. D_{TV}

3.1 Learning to Generate Training-Like Data

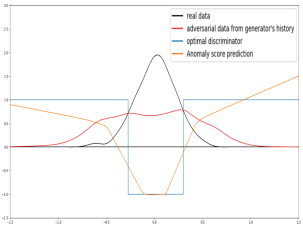
As mentioned, a GAN-based adversarial strategy is followed. Let us recall that the GAN strategy [33] is based on a game theory scenario between two networks, the generator and the discriminator, having adversarial objectives. The generator maps a noise vector (of density $\mathbb{P}_Z$) from the latent space to the image space trying to trick the discriminator, while the discriminator receives either a generated or a real image and must distinguish between both. This procedure leads the probability distribution of the generated data to be as close as possible, for some distance, to the one of the real data. For the Vanilla GAN [33], the minimized distance is the Jensen-Shannon Divergence, which has arguably bad properties (see Sect. 2 of [49] for details). The authors of [49] introduced the idea of minimizing the Wasserstein-1 distance (denoted as $\mathbb{W}_1$) instead. They proved several of its nice properties, including its continuity and differentiability almost everywhere (under certain hypotheses). The Wasserstein-1 distance can be computed with the Kantorovich duality property: if $\mathbb{P}_1$ and $\mathbb{P}_2$ are two probability distributions, then

$$\mathbb{W}_1(\mathbb{P}_1, \mathbb{P}_2) = \sup_{D \in \mathcal{D}} \mathbb{E}_{x \sim \mathbb{P}_1}[D(x)] - \mathbb{E}_{y \sim \mathbb{P}_2}[D(y)], \tag{1}$$

where $\mathcal{D}$ is set of 1-Lipschitz functions, i.e., in the notations of [50], the set of c-convex functions for the cost function $c(x, y) = \mid x - y \mid$. Let G and D



(a) Empirical generated distributions; color from blue to red indicates the progression of training.



(b) Comparison of $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$ and profile of optimal D^*_{TV} and trained D_{TV}.

Fig. 3. Method illustration on a toy one-dimensional dataset of points sampled from a normal law.

be the generator and the discriminator learned by optimizing the adversarial Wasserstein GAN loss (WGAN),

$$\inf_{G} \sup_{D \in \mathcal{D}} \mathbb{E}_{x \sim \mathbb{P}_G} [D(x)] - \mathbb{E}_{x \sim \mathbb{P}_{\text{data}}} [D(x)] . \tag{2}$$

Notice that the optimal dual variable D^* obtained from the optimization of (2) will be negative on real data samples and positive on generated ones. In this paper, we use the learning strategy of [51] which is based on approximating the class $\mathcal{D}$ by neural networks D subject to a gradient penalty (forcing the L^2 norm of the gradient of the discriminator with respect to its input to be close to 1). The choice of WGAN instead of other GAN losses favours nice properties such as avoiding vanishing gradients and mode collapse, and achieves more stable training.

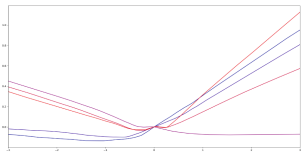
3.2 Generator's History Probability Distribution

Let us start by presenting the underlying idea. During training (equation (2) with algorithm of [51]), the discriminator D will indicate regions that may contain real data, and G learns to produce samples in that zone. If these zones do not contain real data, then the discriminator will act as a critic and indicate it to the generator and point at other regions. This way, screenshots of the generator during training keep track of data points surrounding the real data manifold. In this paper, we merge the screenshots of the generator during training to form:

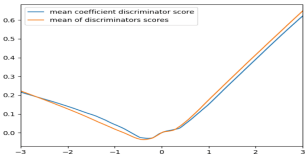
$$\mathbb{P}_{G_{\text{hist}}} \triangleq \int_{\alpha}^{n_{\text{epochs}}} c \cdot G_t(\mathbb{P}_Z) \cdot e^{-\beta t} dt, \tag{3}$$

(see Fig. 2) where G_t denotes the state of the generator at training time t and $\mathbb{P}_Z$ is the latent space distribution (parameters α and β are discussed in Sect. 4). As a weighted mean of training generated distributions, we may assume by construction that $\mathbb{P}_{G_{\text{hist}}}$ covers $\mathbb{P}_{\text{data}}$. That is,

$$\textbf{Hypothesis}: \quad \text{supp}\left(\mathbb{P}_{\text{data}}\right) \subset \text{supp}\left(\mathbb{P}_{G_{\text{hist}}}\right). \tag{4}$$



(a) Discriminator score output during training (from blue to red).



(b) Average outputs of discriminators versus output of a average coefficient discriminator.

Fig. 4. Justification of D_{TV} initialization on the toy example.

To illustrate our hypothesis and our whole method, we present a proof of concept by creating a toy one-dimensional dataset of points sampled from the normal law. We then train the WGAN, with the generator initialized with an offset so that it does not match training data. As previously explained (details in Sect. 4) we save the states of the generator. Figure 3(a) displays empirical generated distributions of some of these states.

In order to satisfy Hypothesis (4), we use momentum based optimizers, so that $\mathbb{P}_G$ oscillates around $\mathbb{P}_{\text{data}}$ (see Fig. 3(a)), making the support of $\mathbb{P}_{G_{\text{hist}}}$ cover the one of $\mathbb{P}_{\text{data}}$ better (see Fig. 3(b), where $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$ are, respectively, plotted in black and red).

This empirically confirms the hypothesis in this toy case.

3.3 Training Our Anomaly Detector D_{TV}

As announced above, we will compute the Total Variation distance between $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$. Let us explain how and why. Firstly, we recall the Total Variation distance definition:

$$\delta(\mathbb{P}_1, \mathbb{P}_2) = \sup_{A \text{ Borel subset}} |\mathbb{P}_1(A) - \mathbb{P}_2(A)|\,, \tag{5}$$

which represents the choice $c(x, y) = 1_{x \neq y}$ in the optimal transport problem, as stated in [50] (where 1_A denotes the indicator function of a set A, as usual). As noticed by several authors (see, *e.g.*, [49]), the topology induced by the Total Variation distance is stronger that the one induced by the Wasserstein-1 distance. Let us remark that $\delta(\mathbb{P}_1, \mathbb{P}_2) = \frac{1}{2}\|\mathbb{P}_1 - \mathbb{P}_2\|_{TV}$, where $\|\cdot\|_{TV}$ denotes the Total Variation norm. The Kantorovich duality yields:

$$2\ \delta(\mathbb{P}_1, \mathbb{P}_2) = \sup_{-1 \leq D \leq 1} \left(\mathbb{E}_{x \sim \mathbb{P}_1}[D(x)] - \mathbb{E}_{y \sim \mathbb{P}_2}[D(y)]\right). \tag{6}$$

From this Eq. (6), we infer our ideal training objective:

$$\sup_{-1 \leq D \leq 1} \mathbb{E}_{x \sim \mathbb{P}_{G_{\text{hist}}}}[D(x)] - \mathbb{E}_{x \sim \mathbb{P}_{\text{data}}}[D(x)]\,. \tag{7}$$

Let us notice that the optimal state D^* in (6) is completely understood: Paraphrasing [49], take $\mu = \mathbb{P}_1 - \mathbb{P}_2$, which is a signed measure, and (P, N) its Hahn

decomposition ($P = \{d\mathbb{P}_1 > d\mathbb{P}_2\}$). Then, we can define $D^* := 1_P - 1_N$, we have $-1 \leq D^* \leq 1$, and

$$\begin{aligned} E_{x\sim\mathbb{P}_1}\left[D^*(x)\right] - \mathbb{E}_{x\sim\mathbb{P}_2}\left[D^*(x)\right] &= \int D^* \mathrm{d}\mu \\ &= \mu(P) - \mu(N) \\ &= \|\mu\|_{TV} \\ &= 2\ \delta\left(\mathbb{P}_1, \mathbb{P}_2\right) \end{aligned} \tag{8}$$

which closes the duality gap with the Kantorovitch optimal transport primal problem, hence the optimality of the dual variable D^*.

Now, we can approximate the Total Variation distance between $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$ by optimizing (7) over D_{TV}, our neural network approximation of the dual variable D.

Several authors (*e.g.*, [36]) have pointed out that the output of a discriminator obtained in the framework of adversarial training is not fitted for anomaly detection. Nevertheless, notice that our discriminator D_{TV} deals with two fixed distributions, $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$. Here, the purpose of computing Total Variation distance is only used to reach the optimal D^*_{TV} in this well-posed problem, assuming Hypothesis (4). D_{TV} should converge to $D^*_{TV} = 1_P - 1_N$ where (P, N) is the Hahn decomposition of $d\mathbb{P}_{G_{\text{hist}}} - d\mathbb{P}_{\text{data}}$ (see Fig. 3(b), blue curve). Importantly, we hope that thanks to the structure of the data ($\mathbb{P}_{G_{\text{hist}}}$ covering $\mathbb{P}_{\text{data}}$), D_{TV} will be able to generalize high anomaly scores on unseen data. Again, this seems to hold true in our simple case: The orange curve in Fig. 3 keeps increasing outside of $\text{supp}(\mathbb{P}_{G_{\text{hist}}})$.

To avoid vanishing gradient issues, we enforce the 'boundedness' condition on D_{TV} not by a *tanh* non-linearity (for instance), but by applying a smooth loss (weighted by $\lambda > 0$) to its output:

$$\lambda \cdot d(D_{TV}(x), [-1, 1])^2 \tag{9}$$

where $d(v, [-1, 1])$ denotes the distance of a real value $v \in \mathbb{R}$ to the set $[-1, 1]$.

Our final training loss, to be minimized, reads:

$$\begin{aligned} \mathcal{L}(D) \ =& \mathbb{E}_{x\sim\mathbb{P}_{\text{data}}}\left[D(x)\right] - \mathbb{E}_{x\sim\mathbb{P}_{G_{\text{hist}}}}\left[D(x)\right] \\ &+ \lambda\, \mathbb{E}_{x\sim\frac{\mathbb{P}_{\text{data}}+\mathbb{P}_{G_{\text{hist}}}}{2}}\left[d(D(x), [-1, 1])^2\right] \end{aligned} \tag{10}$$

As proved in the Appendix, the optimal D for this problem is $D^* = D^*_{TV} + \Delta^*$, with

$$\Delta^*(x) = \frac{d\mathbb{P}_{G_{\text{hist}}}(x) - d\mathbb{P}_{\text{data}}(x)}{\lambda(d\mathbb{P}_{G_{\text{hist}}}(x) + d\mathbb{P}_{\text{data}}(x))} \tag{11}$$

and the minimum loss is

$$-2 \cdot \delta(\mathbb{P}_{\text{data}}, \mathbb{P}_{G_{\text{hist}}}) - \frac{1}{2\lambda} \int \frac{(d\mathbb{P}_{G_{\text{hist}}}(x) - d\mathbb{P}_{\text{data}}(x))^2}{(d\mathbb{P}_{G_{\text{hist}}}(x) + d\mathbb{P}_{\text{data}}(x))} dx. \tag{12}$$

By letting $\lambda \longrightarrow \infty$, the second term in (10) becomes an infinite well regularization term which enforces $-1 \leq D \leq 1$, approaching the solution of (7). This explains why the second term in (12) vanishes when $\lambda \longrightarrow \infty$. In practice, for better results, we allow a small trade-off between the two objectives with $\lambda = 10$.

From now on, we will use the output of D_{TV} as anomaly score. In a nutshell, our method should fully discriminate regions where data points are more dense than synthetic anomalous points from $\mathbb{P}_{G_{\text{hist}}}$. This yields a fast feed-forward anomaly detector that ideally assigns -1 to normal data and 1 to anomalous data. Figure 1 shows an example.

Our method also has the advantage of staying true to the training objective, not modifying it as in [41]. Indeed, the authors of [41] implement a similar method but using a non-converged state of the generator as an anomaly generator. In order to achieve this, they add a term to the log loss that prevents the model from converging all the way.

4 Model Architecture and Implementation Details

In this section, the architecture and implementation of each of the three steps detailed in Sect. 3 is described.

4.1 Architecture Description

G uses transpose convolution to upscale the random features, Leaky ReLU nonlinearities and BatchNorm layers. D is a classic Convolutionnal Neural Network for classification, that uses pooling downscaling, Leaky ReLU before passing the obtained features through fully connected layers. Both G and D roughly have 5M parameters. D_{TV} has the same architecture as D.

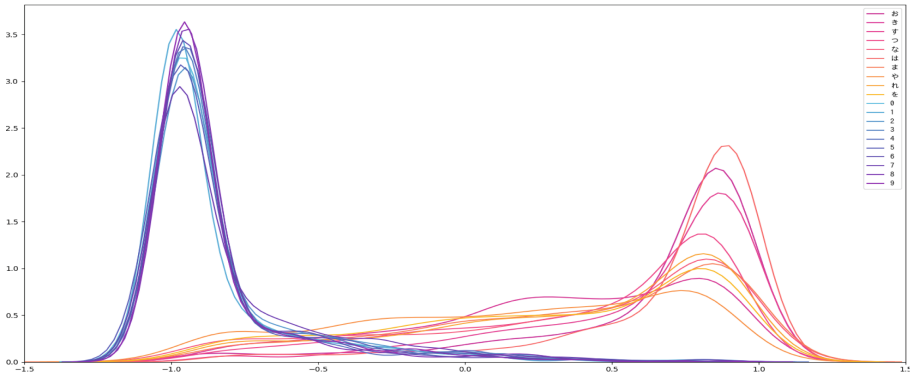
4.2 Learning to Generate Training-like Data

We first train until convergence of G and D according to the WGAN-GP objective of Sect. 3.1 for a total of n_{epochs} epochs, and save the network states at regular intervals (50 times per epoch). We optimize our objective loss using Adam optimizers, with decreasing learning rate initially equal to $5 \cdot 10^{-4}$.

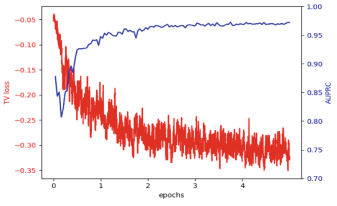
4.3 Generator's History Probability Distribution

As announced in the previous section, if the training process were to be continuous we would arbitrarily define $\mathbb{P}_{G_{\text{hist}}}$ by (3), that is, $\mathbb{P}_{G_{\text{hist}}} = \int_{\alpha}^{n_{\text{epochs}}} c \cdot G_t(\mathbb{P}_Z) \cdot e^{-\beta t} dt$, where c is a normalization constant that makes $\mathbb{P}_{G_{\text{hist}}}$ sum to 1. We avoid the first α epochs to avoid heavily biasing $\mathbb{P}_{G_{\text{hist}}}$ in favour of the initial random state of the generator. The exponential decay gives less importance to higher fidelity samples at the end of the training. In practice, we approximate $\mathbb{P}_{G_{\text{hist}}}$ by sampling data from $\mathbb{P}_{G_t}$ where t is a random variable of density of probability:

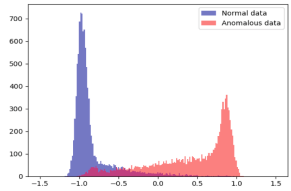
$$c \cdot 1_{[\alpha, n_{\text{epochs}}]} \cdot e^{-\beta t} \tag{13}$$



Approximate density of score distribution among each class of datasets (blue shades: MNIST, red shades: KMNIST).



Red curve: Total Variation distance minimisation. Blue curve: AUPRC of the model during training.



Histograms of anomaly scores for the test splits of MNIST and KMNIST.

One sample from each class of MNIST and KMNIST.

Fig. 5. Method trained on MNIST (normal) and evaluated on KMNIST (anomalous).

4.4 Training Our Anomaly Detector $\mathbf{D_{TV}}$.

D_{TV} is also optimized using Adam algorithm. Since it has the same architecture as the discriminator used in the previous WGAN training, its weights, denoted as $W_{D_{TV}}$, can be initialized as:

$$W_{D_{TV}} = \int_{\alpha}^{n_{\text{epochs}}} c \cdot W_{D_t} \cdot e^{-\beta t} dt, \tag{14}$$

where D_t is the state of the WGAN discriminator at training time t. Let us comment on the reason of such initialization. Figure 4(a) seems to indicate that averaging the discriminators' outputs is a good initialization. The obtained average is indeed shaped as a 'v' centered on the real distribution in our toy example. Figure 4(b) empirically shows that the initialization of the discriminator with

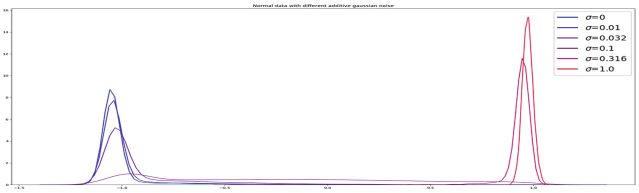


Fig. 6. Method trained on MNIST (normal) and evaluated on modified MNIST images for different levels σ of Gaussian additive noise.

average coefficients is somewhat close to the average of said discriminators. As explored in [52], this Deep Network Interpolation is justified by the strong correlation of the different states of a network during training. To further discuss how good is exactly this initialization, Fig. 7 (blue error bars in the figure) shows a comparison of area under the precision-recall curve (AUPRC) with other methods on the MNIST dataset. The x-axis indicates the MNIST digit chosen as anomalous.

The following experimental cases are tested:

- **Experimental case 1:** The training explained above is implemented.
- **Experimental case 2:** The same process is applied, only modifying $\mathbb{P}_{G_{\text{hist}}}$, corrupting half generated images, by sampling the latent variable with a wider distribution $\mathbb{P}_{Z'}$:
  $\mathbb{P}_{G_{\text{hist}}} = \int_{\alpha}^{n_{\text{epochs}}} c \cdot \frac{G_t(\mathbb{P}_Z)+G_t(\mathbb{P}_{Z'})}{2} \cdot e^{-\beta t} dt.$
  The idea behind this is encouraging $\mathbb{P}_{G_{\text{hist}}}$ to spread its mass further away from $\mathbb{P}_{\text{data}}$.

Figure 1 was computed on experimental case 1 with $n_{\text{epochs}} = 10$, $\alpha = 1$ and $\beta = 5$. Figure 5, 6 and 7 were computed with $n_{\text{epochs}} = 5$, $\alpha = 1$, $\beta = 3$.

5 Experimental Results and Discussion

This section presents quantitative and qualitative experimental results.

Our model behaves as one would expect when presented normal images modified with increasing levels of noise, which is attributing an increasing anomaly score to them. This is illustrated in Fig. 6 where a clear correlation is seen between high values of the standard deviation of the added Gaussian noise and high density of high anomaly scores.

As a sanity check, we take the final state of the generator, trained on MNIST with (2) and [51] algorithm, and verify that our method is able to detect generated sample that do not belong to the normal MNIST distribution. In Fig. 8, we randomly select two latent variable (z_1 and z_2) which are confidently classified as normal, then linearly interpolate all latent variables between them, given by $(1-t)z_1 + tz_2, \forall t \in [0,1]$. Finally, we evaluate the anomaly score of each generated image.

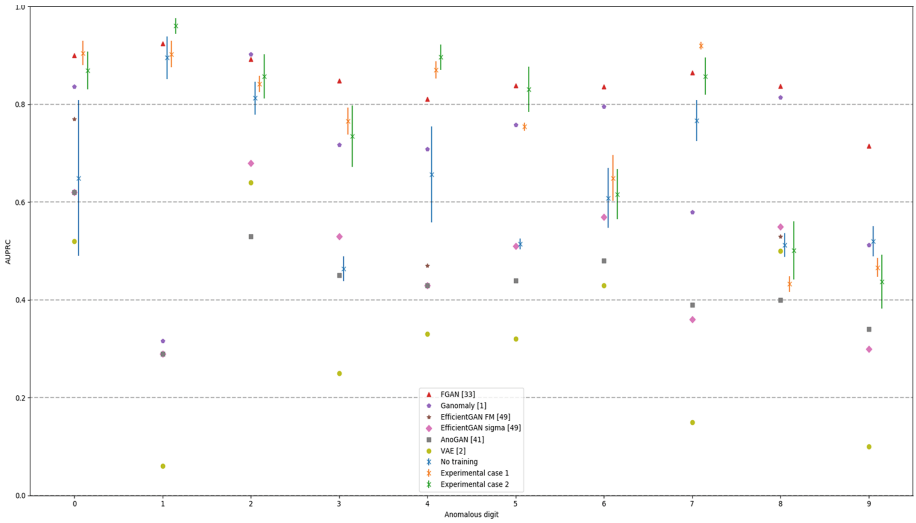


Fig. 7. Comparison of AUPRC with other methods (x-axis denotes the MNIST digit chosen as anomalous).

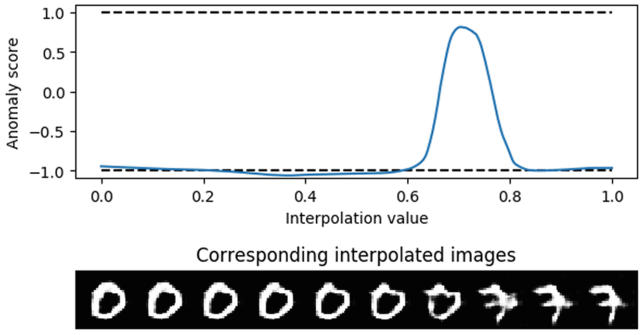


Fig. 8. Method trained on MNIST and evaluating scores on images generated from interpolated latent variables $(1-t)z_1 + tz_2$, for $t \in [0, 1]$.

Finally we check the influence of the number of saves per epoch on the performance of the model. Figure 9 displays the AUPRC of normal data (MNIST) against KMNIST dataset for different values of saving frequency during WGAN training. For low values, the information carried by $\mathbb{P}_{G_{\text{hist}}}$ starts at a 'early stopping of GANs' ([40]) level, and gets richer as the number of saves per epoch increases; hence the increase in AUPRC. We do not have an explanation for the small decay in performance for big values.

Figure 7 compares six state-of-the-art anomaly detection methods with the presented method with both experimental cases and with our D_{TV} initialization (denoted as no training). Apart from a few digits, HistoryAD challenges state of

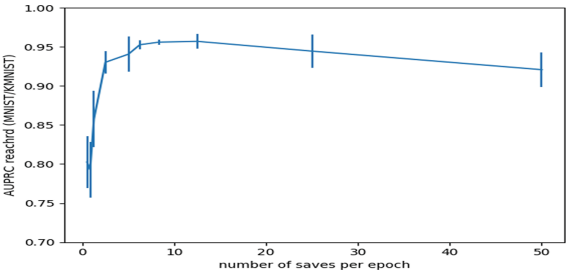


Fig. 9. Influence of the number of checkpoints per training epoch

the art anomaly detection methods. Figure 5 shows the anomaly detection results when our method was trained on MNIST dataset and evaluated on KMNIST. Notice that most of the histogram mass of normal and anomalous data is located around -1 and 1, respectively. This figure empirically proves the robustness of the method to anomalous data structurally close to training data. On the other hand, Fig. 1 shows how well the method performs on structurally different data. Our method was trained on Street View House Numbers [24], and reached high AUPRC results. Both the approximate density and the AUPRC comparison show that the presented method is able to discriminate anomalous from normal data.

6 Conclusions and Future Work

In this paper, we presented our new anomaly detection approach, HistoryAD, and estimated its performance. Unlike many GAN-based methods, we do not try to invert the generator's mapping, but use the rich information of the whole training process, yielding an efficient and general anomaly detector. Further can be done in exploiting the training process of GANs, for instance, using multiple training histories to improve the adversarial complexity of $\mathbb{P}_{G_{\text{hist}}}$.

Appendix

The goal of this appendix is to obtain a solution of the minimization problem

$$\min_D \mathcal{L}(D) \tag{15}$$

where $\mathcal{L}(D)$ is given by (10). Assuming that the probability distributions $\mathbb{P}_{\text{data}}$ and $\mathbb{P}_{G_{\text{hist}}}$ admit densities $d\mathbb{P}_{\text{data}}(x)$ and $d\mathbb{P}_{G_{\text{hist}}}(x)$, respectively, the loss can be written as integral of the point-wise loss l defined below in (17):

$$\mathcal{L}(D) = \int l(D(x))dx \tag{16}$$

where

$$l(D(x)) = (d\mathbb{P}_{\text{data}}(x) - d\mathbb{P}_{G_{\text{hist}}}(x))D(x) + \lambda \frac{d\mathbb{P}_{\text{data}}(x) + d\mathbb{P}_{G_{\text{hist}}}(x)}{2} d(D(x), [-1, 1])^2. \tag{17}$$

Let us recall that $D^*_{TV} = 1_P - 1_N$ where (P, N) is the Hahn decomposition of $d\mathbb{P}_{G_{\text{hist}}} - d\mathbb{P}_{\text{data}}$ (therefore, $\text{sign}(D^*_{TV}) = D^*_{TV}$).

We notice that for all x and for all $\epsilon > 0$,

$$l[(1-\epsilon)D^*_{TV}(x)] \geq (d\mathbb{P}_{\text{data}} - d\mathbb{P}_{G_{\text{hist}}})(x)(1-\epsilon)D^*_{TV}(x) \tag{18}$$

$$> (d\mathbb{P}_{\text{data}} - d\mathbb{P}_{G_{\text{hist}}})(x)D^*_{TV}(x) \tag{19}$$

$$\text{i.e.} > l[D^*_{TV}(x)] \tag{20}$$

Indeed, inequality (18) comes from the positivity of the distance $d(\cdot, [-1, 1])$. On the other hand, inequality (19) comes from the definition of D^*_{TV}. Indeed, if $d\mathbb{P}_{\text{data}}(x) - d\mathbb{P}_{G_{\text{hist}}}(x) < 0$, then $D^*_{TV}(x) = 1$; the other case $d\mathbb{P}_{\text{data}}(x) - d\mathbb{P}_{G_{\text{hist}}}(x) > 0$ gives $D^*_{TV}(x) = -1$. Either way, we obtain that $(d\mathbb{P}_{\text{data}}(x) - d\mathbb{P}_{G_{\text{hist}}}(x))(1-\epsilon)D^*_{TV}(x) > (d\mathbb{P}_{\text{data}}(x) - d\mathbb{P}_{G_{\text{hist}}}(x))D^*_{TV}(x)$. Finally, inequality (20) is obtained from $d(D^*_{TV}(x), [-1, 1]) = 0$.

We can always write a real function D as $D = D^*_{TV} + \Delta$, where Δ is a certain function. We just proved that if $\text{sign}(\Delta(x)) = -\text{sign}(D^*_{TV}(x))$ on a non-negligible set, then D cannot minimize (10), since $D^*_{TV}(x)$ achieves lower value than $D(x)$ on this set.

Hence all minimizer D^* of (10) must be of the form $D^*(x) = (D^*_{TV} + \Delta)(x)$, where $\text{sign}(\Delta) = \text{sign}(D^*_{TV})$ almost everywhere. We can now re-write the point-wise loss formula (17) as

$$l(D(x)) = (d\mathbb{P}_{\text{data}}(x) - d\mathbb{P}_{G_{\text{hist}}}(x)) \cdot (D^*_{TV}(x) + \Delta(x)) \tag{21}$$

$$+ \lambda \frac{d\mathbb{P}_{\text{data}}(x) + d\mathbb{P}_{G_{\text{hist}}}(x)}{2} \Delta(x)^2(x) \tag{22}$$

$$= -2 \cdot \delta(\mathbb{P}_{\text{data}}, \mathbb{P}_{G_{\text{hist}}}) \tag{23}$$

$$+ \int (d\mathbb{P}_{\text{data}} - d\mathbb{P}_{G_{\text{hist}}}) \cdot \Delta + \lambda \frac{d\mathbb{P}_{\text{data}} + d\mathbb{P}_{G_{\text{hist}}}}{2} \Delta^2 \tag{24}$$

Minimizing this point-wise second order equation in Δ, we obtain

$$\Delta^*(x) = \frac{d\mathbb{P}_{G_{\text{hist}}}(x) - d\mathbb{P}_{\text{data}}(x)}{\lambda(d\mathbb{P}_{G_{\text{hist}}}(x) + d\mathbb{P}_{\text{data}}(x))}. \tag{25}$$

Finally, the minimum loss is

$$-2 \cdot \delta(\mathbb{P}_{\text{data}}, \mathbb{P}_{G_{\text{hist}}}) - \frac{1}{2\lambda} \int \frac{(d\mathbb{P}_{G_{\text{hist}}}(x) - d\mathbb{P}_{\text{data}}(x))^2}{(d\mathbb{P}_{G_{\text{hist}}}(x) + d\mathbb{P}_{\text{data}}(x))} dx. \tag{26}$$

References

1. Grosjean, B., Moisan, L.: A-contrario detectability of spots in textured backgrounds. J. Math.Imaging Vis. **33**(3), 313 (2009)
2. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging, pp.146–157. Springer (2017)
3. Prokopetc, K., Bartoli, A.: Slim (slit lamp image mosaicing): handling reflection artifacts. Int. J. Comput. Assist. Radiol. Surg. **12**, 1–10 (2017)
4. Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting fake news: image splice detection via learned self-consistency. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 101–117 (2018)
5. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1053–1061 (2018)
6. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: manipulation tracing network for detection and localization of image forgeries with anomalous features. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
7. Nikoukhah, T., Anger, J., Ehret, T., Colom, M., Morel, J.M., von Gioi, R.G.: Jpeg grid detection based on the number of DCT zeros and its application to automatic and localized forgery detection. In: CVPR, pp. 110–118 (2019)
8. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked RNN framework. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 341–349 (2017)
9. Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S.: Learning regularity in skeleton trajectories for anomaly detection in videos. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
10. Zhong, J.-X., Li, N., Kong, W., Liu, S., Li, T.H., Li, G.: Graph convolutional label noise cleaner: train a plug-and-play action classifier for anomaly detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
11. Nguyen,T.-N., Meunier, J.: Anomaly detection in video sequence with appearance-motion correspondence. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
12. Ionescu, R.T., Khan, F.S., Georgescu, M.-I., Shao, L.: Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
13. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: Temporal cycle-consistency learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1801–1810 (2019)
14. Tout, K., Retraint, F., Cogranne, R.: Automatic vision system for wheel surface inspection and monitoring. In: ASNT Annual Conference 2017, pp. 207–216 (2017)
15. Zontak, M., Cohen, I.: Defect detection in patterned wafers using anisotropic kernels. Mach. Vis. Appl. **21**(2), 129–141 (2010)
16. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVtec AD – a comprehensive real-world dataset for unsupervised anomaly detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019

17. Mishne, G., Cohen, I.: Multiscale anomaly detection using diffusion maps. IEEE J. Sel. Top. Signal Process. **7**(1), 111–123 (2013)
18. Nussberger, A., Grabner, H., Van Gool, L.: Robust aerial object tracking from an airborne platform. IEEE Aerosp. Electron. Syst. Mag. **31**(7), 38–46 (2016)
19. Li, Y., Liu, N., Li, J., Du, M., Hu, X.: Deep structured cross-modal anomaly detection. arXiv preprint arXiv:1908.03848 (2019)
20. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 15 (2009)
21. Pimentel, M.A.F., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. Signal Process. **99**, 215–249 (2014)
22. Ehret, T., Davy, A., Morel, J.-M., Delbracio, M.: Image anomalies: a review and synthesis of detection methods. J. Math. Imaging Vis. **61**, 1–34 (2019)
23. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: a survey. arXiv preprint arXiv:1901.03407 (2019)
24. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
25. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical report, Citeseer (2009)
26. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
27. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, , Fei-Fei,L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
28. Schweizer, S.M., Moura, J.M.F.: Hyperspectral imagery: clutter adaptation in anomaly detection. IEEE Trans. Inf. Theory **46**(5), 1855–1871 (2000)
29. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lect. IE **2**(1), 1–8 (2015)
30. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. CoRR, abs/1703.05921 (2017)
31. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems, pp. 4790–4798 (2016)
32. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv:1312.6114 (2013)
33. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
34. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A.: Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
35. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient GAN-based anomaly detection. arXiv preprint arXiv:1802.06222 (2018)
36. Deecke, L., Vandermeulen, R., Ruff, L., Mandt, S., Kloft, M.: Image anomaly detection with generative adversarial networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 3–17. Springer (2018)
37. Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., Sebe, N.: Abnormal event detection in videos using generative adversarial nets. In: ICIP, pp. 1577–1581. IEEE (2017)

38. Haloui, I., Gupta, J.S., Feuillard, V.: Anomaly detection with Wasserstein GAN. arXiv preprint arXiv:1812.02463 (2018)
39. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: semi-supervised anomaly detection via adversarial training. In: Asian Conference on Computer Vision, pp. 622–637. Springer (2018)
40. Tresp, V., Gu, J., Schubert, M.: Semi-supervised outlier detection using a generative and adversary framework. E& T Int. J. Comput. Inf. Eng. **12**(10), 2018
41. Ngo, C.P., Winarto, A.A., Kou, C.K.L., Park, S., Akram, F., Lee, H.K.: Fence GAN: towards better anomaly detection. CoRR, abs/1904.01209 (2019)
42. Di Mattia, F., Galeone, P., De Simoni, M., Ghelfi, E.: A survey on GANs for anomaly detection. CoRR, abs/1906.11632 (2019)
43. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. CoRR, abs/1605.09782 (2016)
44. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1x1 convolutions. In: Advance Neural Information Processing Systems, pp. 10215–10224 (2018)
45. Choi, H., Jang, E., Alemi, A.A.: Waic, but why? Generative ensembles for robust anomaly detection. arXiv preprint arXiv:1810.01392 (2018)
46. Hendrycks, D., Mazeika, M., Dietterich, T.G.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606 (2018)
47. Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B.: Do deep generative models know what they don't know? arXiv preprint arXiv:1810.09136 (2018)
48. Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J.F., Luque, J.: Input complexity and out-of-distribution detection with likelihood-based generative models. arXiv preprint arXiv:1909.11480 (2019)
49. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
50. Villani, C.: Optimal Transport: Old and New, vol. 338. Springer, Heidelberg (2008)
51. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Advance Neural Information Processing System, pp. 5769–5779 (2017)
52. Wang, X., Yu, K., Dong, C., Tang, X., Loy, C.C.: Deep network interpolation for continuous imagery effect transition. CoRR, abs/1811.10515 (2018)

Deep Transfer Learning Based Web Interfaces for Biology Image Data Classification

Ting Yin[1], Sushil Kumar Plassar[1], Julio C. Ramirez[2], Vipul KaranjKar[1], Joseph G. Lee[2,3], Shreya Balasubramanian[2,3], Carmen Domingo[2], and Ilmi Yoon[1(✉)]

[1] Computer Science Department, San Francisco State University, San Francisco, USA
ilmi@sfsu.edu
[2] Biology Department, San Francisco State University, San Francisco, USA
[3] Dougherty Valley High School, San Ramon, USA

Abstract. Artificial Intelligence (AI) continues to play an integral role in the modernization of the medical and scientific fields. Recent advances in machine learning such as the Convolutional Neural Network (CNN), have demonstrated the ability to recognize complex images with very low error rates using relatively small data set (thousands of images) to become fully trained. Scientists including who are not familiar with programming begin to recognize the need to incorporate machine learning in their research methods to improve the accuracy and the speed of diverse data manipulation without depending on computer scientists. Several tools are developed to serve for these purposes, but such tools are mostly targeting data scientists and often too general or too many options to configure for biologists without machine learning knowledge to get started. We present our work on incorporating Deep Learning into one specific research pipeline that studies how genes work together to regulate muscle formation in the vertebrate frog embryo, *Xenopus laevis*. This research method uses a knockdown approach to diminish the expression of key genes and study how this loss-of-gene function affects the process of muscle formation and differentiation, using mostly fluorescent microscopy techniques which requires time-consuming and challenging visual classifications. We utilized CNN-pretrained transfer learning on the data set with a few different hyper parameters and trained a model with 99% accuracy. Using this experience and discussion with scientists new to machine learning, we developed web interfaces for easy-to-use and complete workflow for scientists to create different classification classes to train, predict and incorporate into their research pipeline.

Keywords: CNN-pretrained transfer learning · Web interface · Frog embryo microscopy · Machine learning

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 777–788, 2021.
https://doi.org/10.1007/978-3-030-55180-3_59

1 Introduction

1.1 Motivation

Deep Learning is a subset of machine learning modeled loosely on the neural pathways of the human brain. *Deep* refers to the multiple layers with nonlinearity between the input and output layers. This enables the algorithm automatically learns complex features from extremely large datasets [1]. Once trained, the algorithms can be applied to new datasets to analyze and derive structures in data that are too large and complex for the human brain to comprehend. Deep learning techniques are particularly well suited to solve problems in data-rich disciplines such as Biology [2].

While Deep Learning has advanced recognition of patterns and understanding of data in various formats, recent advancements in Convolution Neural Network (CNN) achieve super-human level computer vision in natural photo images [3]. Typical human error in image recognition is 5% while the latest CNN models such as VGG19, Inception achieves 3% or lower rates [4]. These achievements are made with advanced Convolution Neural Network (CNN) architecture design, careful selection of hyperparameters, the large and good data set (more than 15 million labeled images) and high computation power for long deep learning training; typical computer science lab will have difficulty in achieving the same results. Fortunately, to promote the advancement of deep learning applications, Google, Oxford research group or other most advanced researchers made their fully trained CNN publicly available through "Transfer Learning" technology [4]. However, these fully trained CNN are trained with objects naturally found from photo images such as dog, cat, car, etc. Scientists or domain experts have started to utilize these fully-trained-CNN to classify their scientific data with small additional training process (called fine-tuning).

Transfer Learning technology has been applied to various scientific data sets. The advanced LIGO (Laser Interferometer Gravitational Wave Observatory) detectors has enabled the detection of multiple gravitational wave signals, establishing gravitational wave astrophysics as an active field of research. While these detectors mitigates the effect of most types of noise, advanced LIGO data streams are contaminated by numerous artifacts known as glitches—non-Gaussian noise transients with complex morphologies. Given their high rate of occurrence, glitches can lead to false coincident detections, obscure and even mimic true gravitational wave signals. Therefore, successfully characterizing and removing glitches from advanced LIGO data is of utmost importance. Deep Transfer Learning has been effectively applied to achieve 98% accuracy in glitch classification and removal [5].

Automated recognition and classification of bacteria species from microscopic images have significant importance in clinical microbiology. Bacteria classification is usually carried out manually by biologists using different shapes and morphologic characteristics of bacteria species. The manual taxonomy of bacteria types from microscopy images is time-consuming and a challenging task for even experienced biologists. An automated deep learning based classification approach has been proposed to classify bacterial images into different categories. The ResNet-50 pre-trained CNN architecture has been used to classify digital bacteria images into 33 categories using the transfer learning technique with an average classification accuracy of 99.2% [6].

Google has been researching on automated detection of diabetic retinopathy and diabetic macular edema in retinal fundus photographs [7]. Diabetic retinopathy alone affects about 40 to 45% of people with diabetes, and it often leads to severe vision loss. People with diabetes have a 10% risk of developing diabetic macular edema. Google used transfer learning using Inception-v3. The accuracy and results were validated by a panel of 54 US licensed ophthalmologists and ophthalmology senior residents. After additional serious validations, Google now moved this product from research to clinics [7].

We apply similar transfer learning technique to microscopy images of muscle formation of vertebrate frog embryo. The significance and the nature of the biology research are discussed in Sect. 1.2. Currently microscopy image analysis is carried out manually by biologists and the process is time-consuming, error-prone and challenging. In this paper, we present our experiments with transfer learning technology and the process and progression of accuracy. Initially computer scientists and biologists work together to establish such pipeline. Through the process, we learned the need of easy and intuitive pipeline that any biologists can incorporate machine learning into their research methodology without involving computer scientists. Classification needs change from one focus to another, for example, binary classification of control and mutant to classification of several classes of control, subtle mild mutants and severe mutants. We also found that while various ranges of scientific classifications go through the same or similar pipeline, domain experts (Biologists in our case) without good coding skills or machine learning knowledge have hard time incorporating transfer learning into their research methodology. Especially the process of data cleaning, classification, training, testing and correlating the prediction results tend to be repetitive and confusing, so intuitive tracking of such tasks become important. Google's AutoML [8], NVIDIA Digits [9] and transfer learning with Neural AutoML [4] are developed to provide such service to domain experts. However, they are designed to solve all generic Machine Learning problems with focus on model training and predictions, but not the complete research workflow and too many options for model training can still daunt the biologists who especially don't have programming knowledge. In this study, we develop and present generic classification transfer learning workflow through pipeline web interfaces.

1.2 Biology Research

One of the key questions underlying the research area in stem cell biology is to determine how a cell becomes committed to a particular lineage (i.e., muscle cell) from a population of undifferentiated cells. The long-term goal of the Domingo laboratory is to better understand how genes work together to form, differentiate and maintain the muscle lineage in the vertebrate frog embryo, *Xenopus laevis*. One striking organizational feature of all vertebrate embryos is their segmented body plan. In fact, the diversity in shape, form and size of all vertebrates is ultimately derived from the timely modulation of segmentation in the early embryo. Segmentation of the vertebrate body begins with the partitioning of blocks of the presomitic mesoderm (PSM) called **somites** [10]. Somites form on either side of the neural tube and notochord and their somitic derivatives will differentiate into the entire skeletal musculature system, connective tissue, blood vessels and dermis [11]. In *Xenopus*, muscle cell progenitors in the nascent somite will go through a series of complex morphogenic and cellular steps in the PSM [10]. Using a

lineage mapping approach, we have demonstrated that muscle cell progenitors in the PSM slowly elongate and increase F-actin protrusions before progressively forming intersomitic boundaries. The final steps in somite formation involve a complex 90-degree cell rotation that causes elongated muscle cells to become aligned parallel to the notochord (Fig. 1) [10]. All these steps are repeated every 50 min until all 45 somites are laid out along the anterior- posterior axis of the *Xenopus* embryo [11].

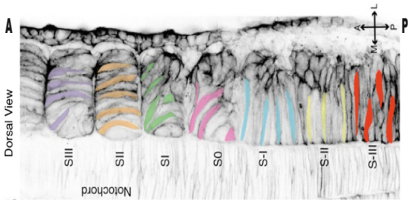


Fig. 1. Formation of somites in *Xenopus laevis*

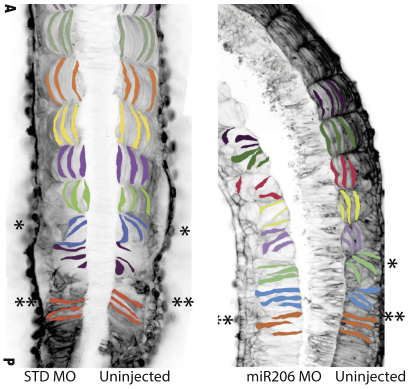


Fig. 2. Muscle cell rotation is disrupted in miR-206 knockdown embryos.

To identify important signaling pathways that control somite formation and differentiation, the Domingo laboratory has used a knockdown approach that can lower the expression of key genes in the developing embryo. We then use confocal microscopy to understand how the disruption of a particular gene impacts cell shape, movement and expression of other key genes during somite formation. Using this approach, we have determined that a gene coding for a small RNA molecule, miRNA-206, is a key factor in muscle formation and differentiation in the frog embryo [12]. Knockdown of miRNA-206 via the use of a morpholino, which binds and sequesters miR-206 from its function, disrupted cell rotation and morphogenesis of muscle cells during somite formation compared to control embryos (Fig. 2). In addition, miR-206 morphant embryos show lower levels of muscle markers (12101) and abnormal deposition of extracellular matrix proteins (Fibronectin). As a result, somites do not segment properly and muscle cell fibers failed to differentiate normally compared to controls (Fig. 3, 4). One big concern in this work is that subtle changes in fluorescent signal or small structural abnormalities within cells go often undetected in our analysis. Therefore, our published work requires the tedious and time-consuming analysis of hundreds of single optical images (Fig. 2, 3 and 4) before we can establish whether a particular miRNA plays a major role in skeletal muscle formation and differentiation. Figure 5 shows typical images in our data set.

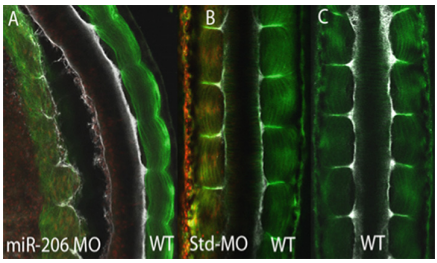


Fig. 3. Muscle formation and differentiation are disrupted in miR-206 knockdown embryos.

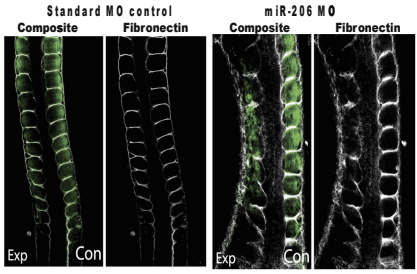


Fig. 4. Impact of miR-206 levels on the distribution fibronectin, an extracellular matrix marker, during somite formation & differentiation.

Fig. 5. Confocal microscopy produced LSM format images. Most of images in our data set are black and white. A and B are typical control images and C and D are strong mutant images.

2 Deep Learning and Transfer Learning Background

Recent success of transfer learning applications is presented in Sect. 1. Here we discuss the brief history of the transfer learning development. ImageNet is a project aimed at labeling and categorizing images for computer vision research into approximately 22,000 different object categories. The name also refers to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The goal of the ILSVRC is to train a model that can accurately identify an input image in 1,000 separate categories [14]. Models are trained on approximately 1.2 million training images with an additional 50,000 validation images and 100,000 test images. The 1,000 image groups represent object categories that people encounter in their daily lives, such as dogs, cats, everyday items, different automobiles, and so on. In 2012, a deep convolution neural net called Alex Net achieved a 16% error rate which was significant leap compared to other approaches in previous years [3, 13]. A Convolutional Neural Network (CNN) is a Deep Learning algorithm which takes an input image and assigns importance (learnable weights and biases) to various aspects of the image allowing it to distinguish images of one category

from another. A CNN's architecture is similar to that of the human brain's Neurons communication pattern and was inspired by the Visual Cortex organization.

Over the years many researchers have participated in these competitions with various CNN architectures and in 2017, 29 out of 38 competing teams had accuracy greater than 95% [3]. Typical human error in image recognition is 5% while the latest ML models such as VGG19, Inception achieves 3% or lower rates. These achievements are made with advanced Convolution Neural Network (CNN) architecture design, careful selection of hyperparameters, the large and good data set (more than 15 million labeled images) and high computation power for long deep learning training. The research organizations that develop models for this competition and do well often release their final model under a permissive license for reuse. These models can take days or weeks to train on modern hardware. Fortunately, to promote the advancement of deep learning applications, Google, Oxford research group or other most advanced researchers made their fully trained CNN publicly available through "Transfer Learning" technology [4]. This approach is effective because the images were trained on a large corpus of photographs and require the model to make predictions on a relatively large number of classes, in turn, requiring that the model efficiently learn to extract features from photographs in order to perform well on the problem. Therefore, transfer learning is similar to human learning process; human who has a good sight for natural objects receives additional training for professional classification, then the person will become good at classifications on the trained classes. Transfer Learning starts with a pre-trained CNN to enable classification of specific classes with new data sets on those specific classes. This new data set may be much smaller than typical machine learning algorithms need to be trained due to its pre-trained nature. This fits to many research lab and scientific data classification very well as such research lab may not have large data set for each class that they want to study, man power to label such data sets nor heavy computation power needed to train from scratch.

3 Dataset and Classification Results

Our data collection process is shown in Fig. 5. At the end of the process, confocal microscopy produces images in LSM format which in itself is a combination of multiple Z stacks and channels based on the number of antibodies used. These images are extracted via Fiji software which is an open-source image processing package based on ImageJ. The existing image extraction process is lengthy and takes time and effort to segregate and store images with respect to their name and type of antibody used. To train a model properly, cleaning or organizing the data set is highly important as the model learns from the data and the domain knowledge is essential in doing the task. Some data sets are in gray area in classification and handling such data sets need to be delicate and often challenging. These cumbersome processes well explain that generating large labeled data set is very challenging and expensive in typical biology lab (Fig. 6).

The dataset provided was uploaded to the web interface experiment by experiment (more details will be explained in next section). Each image within the experiment was cropped into multiple segments as the embryo image is very long while CNN usually takes square sized images. CNN can work with any resolution image by internally

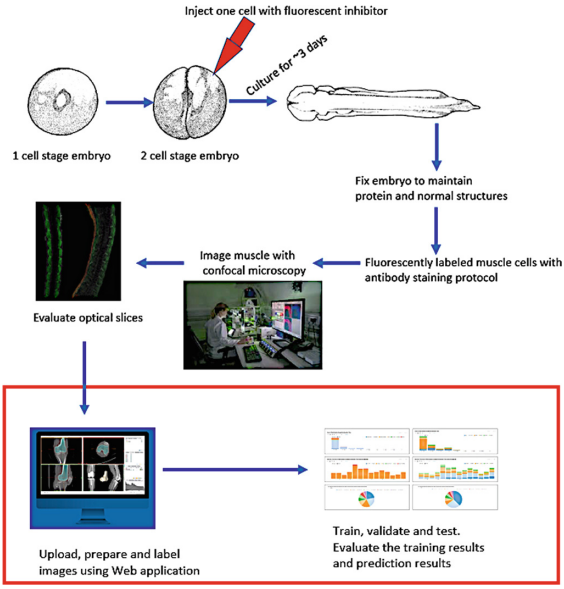


Fig. 6. Full process pipeline integrated with Deep Learning process

resizing them, but we wanted to produce more images with details. The given image works well with three divided crops. Then, every image was converted into three cropped images using our web interface developed in next section. The idea behind this was, the three-segment could individually focus on different parts of the embryo regions and capture their characteristics to classify them efficiently, which was not easily captured when the image is considered as one single image. Other biological images will have similar issues, so we developed efficient group cropping interface.

We started machine learning process with 402 control images and 369 mutant images. From them, we saved 25% as testing data (Control: 99 images (25% of 402), Mutant: 93 images (25% of 369)). After trying many different hyperparameter combinations, we got 87% accuracy - ((True Positive + True Negative)/Total) * 100 = 87.33%. 87% accuracy was a decent starter, but there was lots of room to improve. We have tried several different combinations of hyperparameters. After that, we started by analyzing the dataset, to understand what went wrong which ultimately helped in improving the accuracy [15]. There are many mild difference between the mutant and control images which were troublesome to differentiate even for the Biology Researchers as well. Due to this reason, the ML Model was getting confused in differentiating between those two categories especially considering that our data set is relatively small. Some of the mild mutants were predicted as controls as they possess their majority of the characteristics associated with the controls and minority of the characteristics associated with the mutants. The ML Model find these mild mutants were mostly aligning with control, so it was predicted as control, but it was incorrect even though it makes more sense. This cleaning of the training data set boosted the accuracy to 92.59%.

Further modifications were made to the datasets in order to ensure the datasets in all categories were balanced. Previously the control dataset had 402 items, and the mutant dataset had 369 items resulting in an unbalanced dataset. A balanced dataset would have the same number of examples for each class (same number of controls and mutants). Without balancing our input dataset, the model learn to predict whichever category has the most samples. The remedy to avoid this issue is to either pass class weights to the model so that it can measure errors appropriately, or to balance the samples by trimming the larger set until both sets are the same size. The approach taken to fix the unbalanced dataset was to trim the larger dataset, in this case the mutant dataset, until it matched the smaller control dataset. The accuracy is bumped to 96.8%

After these learning process, our biology research team generated a new data set. And the average of the latest data set is 98%.

4 Complete Workflow Web Interfaces

From the experiences above, we have found the strong need to web-based interfaces where biology researchers can easily create the classes for various classification needs, upload images that automatically label, group cropping if needed, run predictions, validate the results (that can collect more data sets for additional training) and see the statistics of predictions without involving computer scientists. At the same time, many biologists are concerned about data security, so they may not want to upload their data to publicly available machine learning cloud services. So, we make these web interfaces available as code repository, so they can create and own their own private cloud computing. The web interfaces focuses on the workflow of Deep Transfer Learning classification training and prediction.

4.1 Training

Machine learning workflow is split into training (Fig. 7, 8, 9, 10 and 11) and prediction (Fig. 12, 13, 14, 15 and 16) phase. A Machine learning model needs to be well trained to be useful in prediction. If a model is sufficiently trained for a desired accuracy, the model can be shipped for prediction process while the model can be periodically retrained and evolve. Once data is organized into classes, data sets need to be split into training (& validation) and test data set. Figure 8 shows this process of creating classes and a button to upload images per corresponding classes.

ML typically has three types of datasets: training, testing, and validation [15]. The first dataset is the training dataset which is the largest of the three. It is a set of data used for learning that is to fit the parameters of the model. The model sees and learns from this data. The validation dataset is the data used to provide an unbiased evaluation of a model fit during the training set while tuning the model hyperparameters. This set can be used for regularization by early stopping, i.e., stop training when the error on the validation set increases [15]. Finally, the test dataset is used to provide an unbiased evaluation of a final model on the training dataset.

In ML, hyperparameters are type of values that control the learning/training process. The hyperparameters include the number of epochs (how many iterations of the

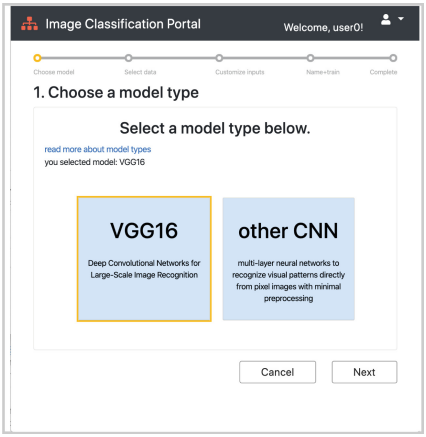


Fig. 7. Model type selection

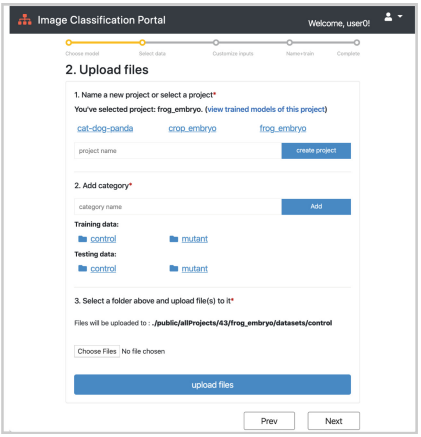


Fig. 8. Upload files

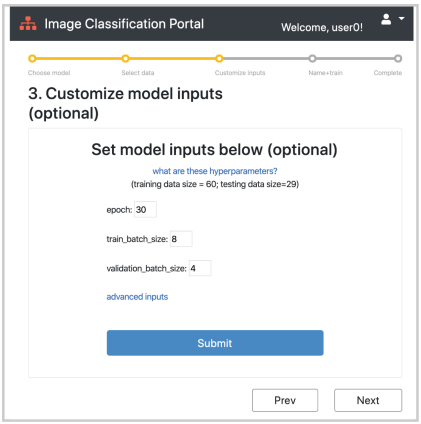


Fig. 9. Model hyper-parameters

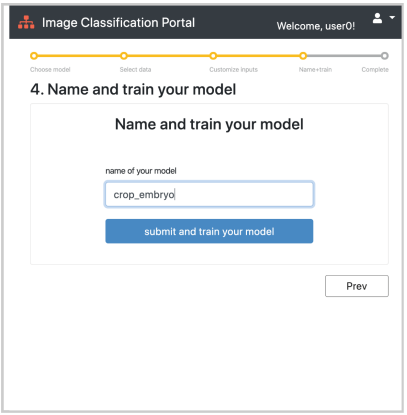


Fig. 10. Name & train the model

complete data set), batch size (size of batch for one learning step), etc. There is no predefined number to be set for the hyperparameter, but they are tuned to discover the parameters of the model that has the best performance. This is a crucial step as using the right hyperparameters will result in skillful prediction and it depends on the data sets. We set default values for these and plan to put guidance or suggestions on how to increase or decrease by observing the training results. Advance button also allows tuning more advanced options in selecting optimizers, learning rates, regularization, etc. Very often, the training process takes longer time than other interactive tasks. We send email notification, so user can be notified with the training log.

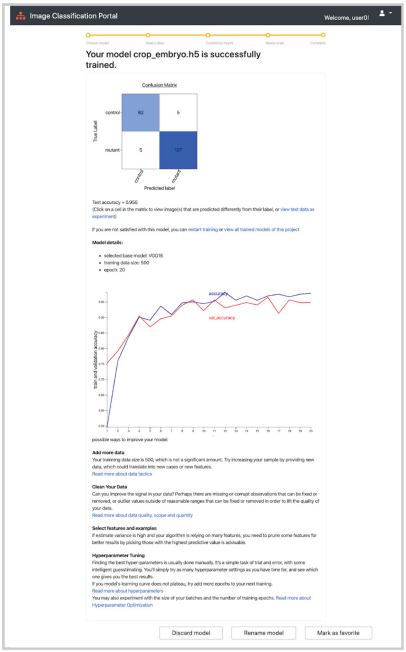


Fig. 11. Model training logs

4.2 Prediction

After successful training of models and high accuracy result of test data set, scientists would like to go through good period of validation of new prediction results. Often multiple lab staffs or assistants will participate in the validation process, too. Therefore, tracking which experiment and which images predictions have been validated is important. Each experiment has its own tags such as antibiotic materials or florescent materials used in the experiment setting. To make the validation of prediction or review process user-friendly and easy, we produce statistics of the batch of images from a given experiment as well as easy to use and effective interface to validate individual image. Figure 12 is the interface where scientists validate individual image on the prediction results by toggling the yes/no buttons. The validated image gets highlighted in a different color to keep track of what all was validated so far. Scientists can also select all the images in one go as yes/no and confirm the predicted results.

Once the image is validated, it moves to the next tab i.e. Fig. 13 where users can see which image was wrongly classified. Clicking on the image name pops out a full image preview as shown in Fig. 14.

To help the scientists keep track of all the predictions together, a simplistic user interface is generated which provides a summary report as shown in Fig. 15. Additionally, one can also view summary of all the test data results conducted during the training phase as displayed in Fig. 16. After clicking on an experiment name in the summary view, scientist can explore through further details related to that experiment.

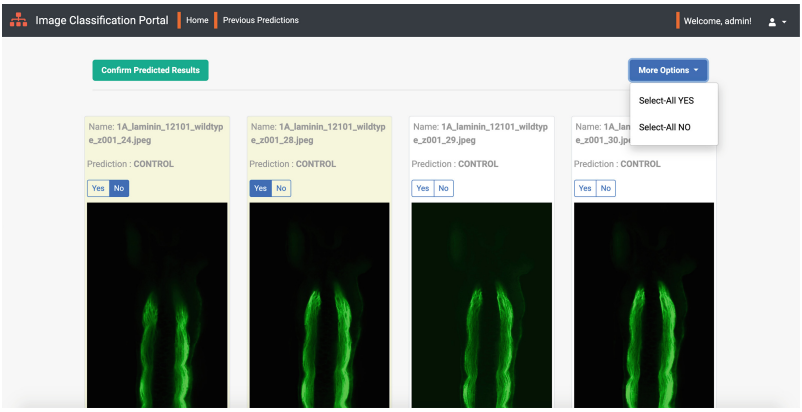


Fig. 12. Validating result predicted by model

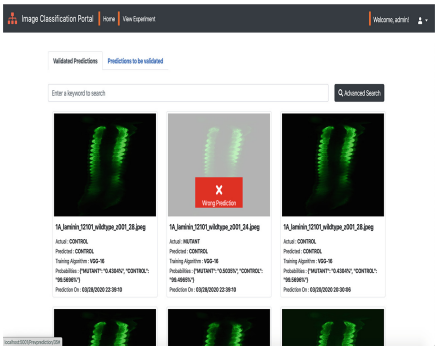


Fig. 13. Classification of right/wrong images.

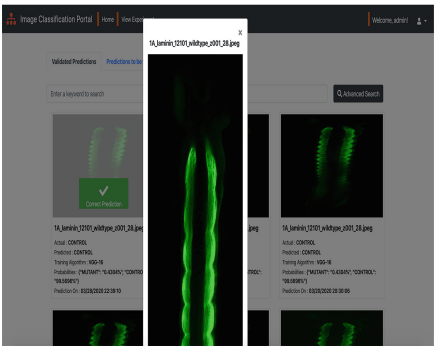


Fig. 14. Enlarging image to preview original

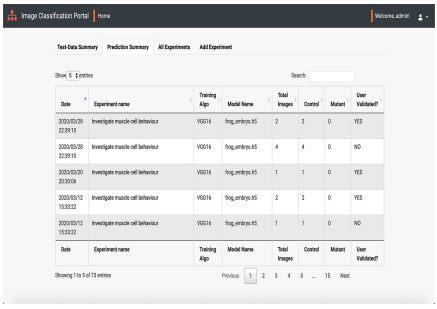


Fig. 15. Prediction data summary

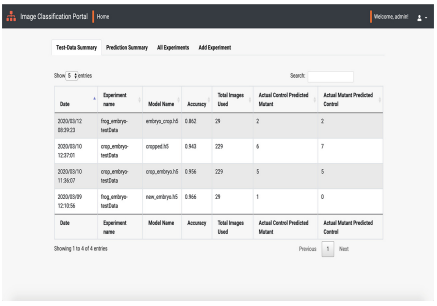


Fig. 16. Test data summary

5 Conclusion

Deep Learning, especially Deep Transfer Learning opens new doors to scientists who desires to detect patterns from data or classify classes with high accuracy and speed in their research pipeline. However, Deep Transfer Learning or Machine Learning pipeline seems daunting or having very heavy learning curve to many scientists who are not familiar to programming. A few machine learning Cloud web sites tend to solve all categories of problems, so scientists without prior experience get lost easily. We present easy to use web interfaces for Deep Transfer Learning Classification for scientists who are new to machine learning.

References

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). http://www.deeplearningbook.org
2. Ching, T.: Opportunities and obstacles for deep learning in biology and medicine. J. Roy. Soc. Int. **15**(141) (2018). http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5938574/pdf/
3. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
4. Wong, C., Houlsby, N., Lu, Y., Gesmundo, A.: Transfer learning with neural automl (2018)
5. George, D., Shen, H., Huerta, E.: Classification and unsupervised clustering of ligo data with deep transfer learning. Phys. Rev. D. **97**(10), e101501 (2018). https://doi.org/10.1103/PhysRevD.97.101501
6. Talo, M.: An automated deep learning approach for bacterial image classification (2019)
7. Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P.C., Mega, J.L., Webster, D.R.: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama **316**(22), 2402–2410 (2016)
8. Thomas, R.: Google's automl: Cutting through the hype (2018). https://www.fast.ai/2018/07/23/auto-ml-3/
9. Developer, N.: Nvidia digits, June 2019. https://developer.nvidia.com/digits
10. Afonin, B., Ho, M., Gustin, J.K., Meloty-Kapella, C., Domingo, C.R.: Cell behaviors associated with somite segmentation and rotation in Xenopus laevis. Dev. Dyn. **235**(12), 3268–3279 (2006)
11. Sabillo, A., Ramirez, J., Domingo, C.R.: Making muscle: morphogenetic movements and molecular mechanisms of myogenesis in Xenopus laevis. Seminars Cell Dev. Biol. **51**, 80–91 (2016)
12. Vergara, H.M., Ramirez, J., Rosing, T., Nave, C., Blandino, R., Saw, D., Saraf, P., Piexoto, G., Coombes, C., Adams, M., Domingo, C.R.: miR-206 is required for changes in cell adhesion that drive muscle cell morphogenesis in Xenopus laevis. Dev. Biol. **438**(2), 94–110 (2018)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large scale image recognition. arXiv.org (2015). http://search.proquest.com/docview/2081521649/
14. Wikipedia: Imagenet (2019). https://en.wikipedia.org/wiki/ImageNet
15. Developers, G.: Training and test sets: Splitting data—machine learning crash course. https://developers.google.com/machine-learning/crashcourse/training-and-test-sets/splitting-data

Dangerous State Detection in Vehicle Cabin Based on Audiovisual Analysis with Smartphone Sensors

Igor Lashkov(✉), Alexey Kashevnik, and Nikolay Shilov

SPIIRAS, 39, 14 Line, St. Petersburg 199178, Russia
{igla,alexey,nick}@iias.spb.su

Abstract. The paper presents the context-based approach for monitoring in-vehicle driver behavior based on the audiovisual analysis with aid of smartphone sensors, essentially utilizing front-facing camera and microphone. We propose the approach of driver monitoring system focused on recognizing situations whether the driver is drowsy or distracted, and reducing traffic accidents rate by generating context-relevant recommendations and perceiving driver's feedback in a form of requested audio response to certain speech commands given by the smartphone. We efficiently utilize the information about driving behavior and the context to make sure that the driver actually followed the given recommendations that in the result will aid to reduce the probability of traffic accident. For example, audio signal produced by the smartphone's microphone is used to check whether the driver increased or decreased the music volume inside the vehicle cabin. If the driver did not proceed with the recommendations, the driver is prompted to response with the voice command, and in this way, to confirm its alertness to current driving situation.

Keywords: Driver · Dangerous state · Audio-based assistance · Context · Vehicle

1 Introduction

According to the report of European commission, road traffic accidents cause damage to human's health and take lives of more than 25,000 people on the road of EU annually [1]. The development of driver monitoring and assistance systems is becoming popular nowadays, as it reaches the high level of functionality and performance. It is a promising approach for providing in-vehicle driver safety as the early signs of drowsiness or detection can be detected before the traffic accident arises and relevant context-based recommendations can be generated for a driver e.g. take a short break. Thus, for example, the EU has plans to equip new produced cars with driver behavior monitoring systems focused on recognizing the situations when a driver is distracted or drowsy [2]. Moreover, the Volvo has a vision of future with zero road fatalities by installing the driver monitoring cameras inside the vehicle cabin able to determine the distraction and drunk driving states [3].

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 789–799, 2021.
https://doi.org/10.1007/978-3-030-55180-3_60

Intelligent driver assistant systems actively utilize massive set of different sensors, capturing measurements describing driver behavior, behavior of other road participants, and the environment road situation. It should be noted, that existing driver assistant systems are mainly concentrated on the analysis of situations happening ahead of the driver outside the vehicle cabin. Moreover, these kinds of solutions heavily rely on the use the video cameras, radars, and lidars, limited in processing severe weather conditions, visual subject reflections, requirement of certain level of driver privacy, or having a relatively high price for hardware components. In-vehicle cabin driver behavior state may be analyzed whether cues of drowsiness or distraction state are present and can affect the driving performance, and potentially increase the risk of traffic accident occurrence. This information can aid to describe the current driving situation and, therefore, potentially provide increased accuracy and reliability of dangerous situation determination.

The main purpose of the paper is to present the approach for detection of driving dangerous states in vehicle cabin that relies on the audiovisual analysis accomplished by built-in smartphone sensors, essentially backed by the front-facing camera and the microphone. Audio-based data obtained from smartphone will aid to test the feasibility of driver performance assessment in recognizing cases when the driver is sleepy or distracted through analyzing the driver speech responses, and potentially avoid an emergency situation while driving.

There is a research and technical gap in researching and employing the audio-based solutions aimed at providing driver safety while driving. The visual-based analysis for driver behavior monitoring was essentially covered in our previous works [4, 5]. This paper extends our prior works in this field of research.

Our experiments demonstrated the feasibility of driver behavior monitoring system in terms of early warning driver about the probability of road emergency occurrence. The information about dangerous driving state transmitted to the remote dispatcher may aid to call emergency services and, in consequence, inform the fleet dispatcher about these kind of road situations.

The rest of the paper is organized as follows. Section 2 presents a comprehensive related work in the area of audio-based solutions aimed at recognizing whether the person is drowsy, distracted or stressed at the moment and compares the existing projects and solutions. Section 3 describes in detail the reference model of the proposed audio-based approach for dangerous situation determination. The algorithm for dangerous situation determination based on microphone audio signal is presented in Sect. 4. The implementation and evaluation are presented in Sect. 5. Finally, the main results and potential future work are summarized in Conclusion (Sect. 6).

2 Related Work

This section discusses the research made in driver monitoring systems focused on recognizing drowsiness and distraction states, based on the analysis of the audio signals produced inside the vehicle cabin.

The authors in [6] discuss the changes in the content and speech caused by sleep loss. In detail, they studied the effects of sleep loss on the spontaneous generation of words during a verbal word fluency task and the articulation of speech during a vocalized

reading task. Within natural experiments conducted with nine persons of average age 21 from UK, the researches concluded, that there was a reduction in the use of appropriate intonation in the voice after sleep deprivation, with candidates displaying more monotonic or flattened voices. Also, after sleep deprivation there was a deterioration in word generation tasks and a tendency for candidates to become fixed within a dictionary of semantic similar words.

The study [7] presents a framework for recognizing signs of drowsiness based on the analysis of articulation, prosody, and speech-quality-related speech characteristics per audio sample. Acoustic features are extracted during while recognizing speech, speaker, and emotion, including fundamental frequency, intensity, pause patterns, formant bandwidth, format position, and cepstral coefficients. The authors of the paper concluded that acoustic features calculated from read speech and produced in a result of pattern recognition methods, contain noticeable amount of information about speaker's sleepiness level. The authors of the paper note, that support vector machine model showed 86.1% classification accuracy in predicting sleepiness level, that in comparison outperformed other existing models, including Multilayer Perception classifier, 1-Nearest-Neighbor (NN), 2-NN, 3-NN, Decision Tree, Random forest, Naive Bayes, logistic regression. In addition, different speech samples provided by Emotion and stress speech databases (e.g. FAU Aibo Emotion Corpus [8]) could serve as a model in solving task of acoustic sleepiness analysis.

Another study [9] presents a non-invasive method for recognizing precise cues in the voice allowing to characterize the state of the speaker and measure its sleepiness state. Experiments were conducted with aid of patients having a suspicion of excessive daytime sleepiness and required to read six different text in different time of the days. Along with it, the patients filled the Karolinska Sleepiness Scale [10] after they finish to read texts. The audio files recorded during these sessions were divided in segments with length in a range of 50 s to 2 min. Following audio features were directly extracted from each recording to measure patient sleepiness: the duration of voiced parts, the percentage in duration of voiced parts, the duration of vocalic segments, the percentage in duration of vocalic segments. Other features were calculated on each voiced segment to characterize harmonic sounds and include descriptive values (frequency, power, bandwidth) of harmonics and formants; fundamental frequency and intensity; cepstral peak prominence [11] and Harmonics-to-noise ratio [12].

Patent in [13] describes the method for reducing boredom for the driver of the vehicle in the environments with straight roads and lack of traffic. According to the authors of the paper, a bored driver is typically one that has no particular desire to sleep, but minimal demands on their attention capacity lead to a sense of boredom, which may lead to the drowsiness state even without a particular physiological need to sleep. Thus, the boredom state of the driver is a real danger to safe vehicle operation. In case the driver is considered in a boredom state, the driver may be prompted an assistance to eliminate this dangerous behavior. Text-to-speech module may be utilized to ask the driver directly "Are you bored?" with an appropriate response leading to the use of the stimulation device to help the driver avoid boredom. Some examples of stimulation device can be a semi-transparent interactive display showing photos.

Research paper [14] proposes an approach for recognizing sleepiness state in a voice. Considering time frames, following features are extracted for experiments: the total duration of voiced parts, he percentage of duration of voiced parts, total duration of vocal segments; and percentage in duration of vocal segments. At the same time, measurements on the fundamental frequency and intensity curves were considered as following features: mean and variance of fundamental frequency over a voiced segment; slope of the linear approximation of the fundamental frequency over a voiced segment; minimum and maximum of fundamental frequency over a voiced segment; and extend of fundamental frequency values over a voiced segment. During the conducted experiments, the authors concluded that sleepy speakers struggle to produce the same variety of nuances, frequencies, energy and quality of voiced parts same in comparison with non-sleepy person. The authors of the paper selected 23 features for their method to work, reaching the performance in comparison with state-of-the-art projects.

Abnormal driver behavior can be described with aid of emotions exhibiting by the driver at certain moment of time. The author in [15] proposes an emotion-based approach to recognize driver affective states. The authors developed the affective space model using emotional speech data on three different cultures bases. The driver behavior statistics was taken from the Real-time Speech Driving Dataset. The experiments conducted by the authors demonstrate that the proposed approach allows to recognize a set of different driver states, including sleepiness, talking to mobile phone, laughing while driving and normal driving. Based on the results of the experiments, the authors of the study suppose that automakers are able to recognize driver's fatigue and stress level. Other research works are also related to better understanding speech characteristics of the driver [16].

It can be summarized that audio signals, and, in particular, speech signals, observed in vehicle cabin, can serve as a valuable source of information in continuously monitoring driver behavior and recognizing dangerous states through extracting sings of fatigue, sleepiness, emotional and stress level while driving.

3 Reference Model of Driver Monitoring System

We propose the reference model of driver monitoring system shown in Fig. 1. Initially, the interaction of the driver with the system starts with the smartphone, mounted on the windshield of the vehicle inside the cabin. It should be fixed in the way that the front-facing camera of the smartphone is directed towards the driver's face. The first attempt to use the mobile Drive Safely application on the smartphone requires driver to proceed the calibration step, providing the best performance and efficiency for monitoring driver behavior through adapting to its visual features, including posture, head position, eye state, etc. Essentially, we are focused on determining driver's drowsiness and distraction states utilizing the camera-based approach and measurements obtained from built-in smartphone sensors, like accelerometer, gyroscope, magnetometer, etc., that we earlier described in details in our previous works. Based on the results of dangerous behavior analysis, the context-aware recommendations aimed at road accidents prevention, are given for a driver in a form of audio instructions produced through smartphone's microphone.

In general, recommendations can be classified into two groups: the ones, that may be expressed for a driver in a form of same time audio requests to act right inside the

vehicle cabin (e.g. turn on radio or music, cool the car interior, adjust the seat position, sign yourself); and, the others, that require the driver to make a short or a long stop at a gas station, hotel or café, or just pull over and take a nap for 15–20 min. There is an issue, that the drivers being drowsy or tired, are prone to perceive the environment reasonably. Moreover, most vehicle safety technologies aimed at reducing accident rate are tied to visual representation of current road situation. In this way, the use of additional methods and technologies presented by cloud computations, including Voice Activity Detection, Machine learning algorithms and natural language processing, is proposed to overcome these difficulties, understand the real driving intents and better driver assistance inside the vehicle cabin. We propose an audio-based approach to perceive the situation inside the vehicle cabin and recognize abnormal driving behavior. Essentially, the parameters, describing the audio context of driving behavior, that may aid in dangerous state determination, are the following: audio level of loudness, indicating the magnitude of the sounds (beeps, sirens) happening inside or outside the vehicle; the characteristics of the music or radio playing (e.g. genre, sound level); text of the speech produced by the driver; the presence of emotions in human speech (e.g. neutral, calm, happy, sad, angry, disgust, drowsy, etc.); and the time, the driver need to react to the recommendations given by the monitoring system, measured in milliseconds and indicating the delay in driver decision-making. For instance, the use of the listed parameters can aid to describe slow reaction time of the driver resulting in break or steer suddenly and indicate the first signs of drowsiness or distraction states. It should be noted, that the driver may connect smartphone to vehicle infotainment system via wireless connection, e.g. Bluetooth, providing the increased usability and audio perception with the Drive Safely system.

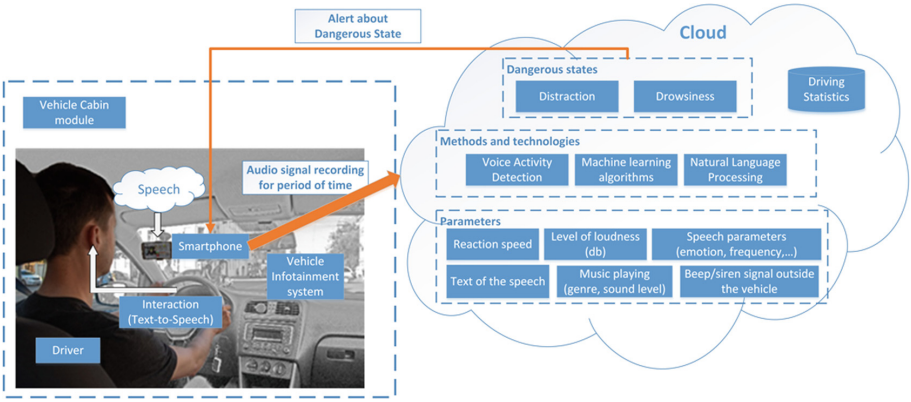


Fig. 1. Reference model of the Drive Safely monitoring system

4 Audiovisual Approach for Dangerous State Determination

The complete scheme of the algorithm for dangerous state determination in driving behavior inside the vehicle cabin is presented in Fig. 2.

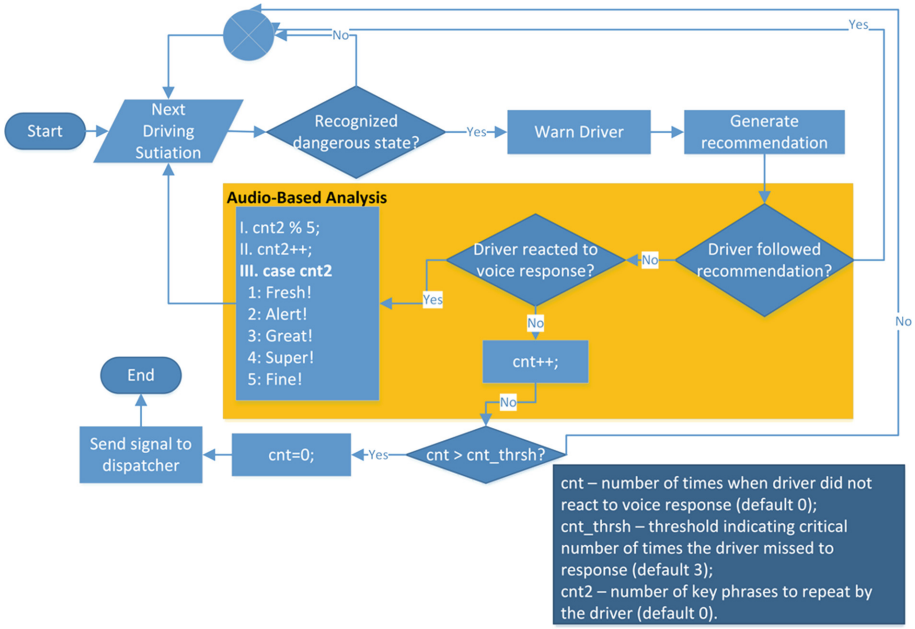


Fig. 2. Algorithm for dangerous state determination in driver behavior

The input for this scheme is presented by the data obtained from smartphone sensors at the moment, and, initially, the 2D image frame captured by front-facing camera is utilized to describe the visual behavior of the driver, its facial features. On first step, the system is oriented on analyzing and processing a batch of video frames containing the visual signals of abnormal driving behavior inside the vehicle cabin, that is observed together with a set of distinct parameters, such as PERCLOS [17], eye-blink rate, number of yawns per minute, left or right head turn or tilting the head up or down. Based on this information, it decides whether the driver is actually drowsy or distracted at the moment and this potentially emergency road situation need in getting more attention by the driver.

On the next step of the algorithm, in case the dangerous driver's state is recognized by the monitoring system, the driver is warned by beep sounds and speech signals produced by text-to-speech module, as well as receives the context-based recommendations aimed at preventing possible traffic accident. In general, recommendations present some set of ordered and clear instructions to follow by the driver right inside the vehicle cabin. Some of them can be voiced as "Cool the car interior", "Talk to passengers", "Take two cups of coffee at the nearest hotel" or "Pull over and take a short nap for 15–20 min".

One of the problems in this case is that the driver does not always completely understands the current driving situation reasonably and making judgments of risks caused by dangerous behavior, drowsiness or distraction states, while driving. Thus, we use the information about driving behavior and the context to make sure that in fact the driver followed the given recommendation. In some cases, we use the sensor data extracted

from smartphone sensors to ensure the driver did not ignore the recommendation. Vehicle location coordinates (longitude, latitude) read from GPS may be analyzed to see whether the driver agreed to stay at the nearest hotel for a rest and he/she heads to the final destination point. At the same time, microphone may be efficiently used by the smartphone application to capture audio signal and to check whether the driver increased or decreased the music volume inside the vehicle cabin, or he/she is driving with passengers. Otherwise, if the driver did not proceed with the recommendations, we ask him/her to response with the voice command earlier defined by the application, and, in this way, to confirm its alertness to current driving situation. Typical examples of possible answers the driver may use are "Fresh", "Alert", "Great", "Super", "Fine", etc. In case, the driver failed to response upon the driver monitoring system request more than *cnt_thrsh* times (experimentally configured to three times), his/her behavior is considered as potentially dangerous, and the information about it is transferred to the emergency dispatcher service for further actions.

Audio-based approach for dangerous situation determination and preventing possible road accidents is presented in Fig. 3. The whole scheme of this approach is proposed in a form of ongoing timeline. It should be noted, that the main source of information is provided by smartphone's microphone, continuously capturing the audio signal inside the vehicle cabin. In order to efficiently operate the audio signal, we use and constantly reuse the temporary sound signal buffer on the smartphone that allows to keep accumulated data for last time, implementing the slide window algorithm, avoiding the extra costs for data processing. Using the cloud platform, we analyze the sound signal, recorded right before the dangerous state occurred for the period of time t_1 in a range from E_0 to E_1 and during the interval the driver got the warnings about emergency situations, recommendations and its reaction to system speech-based request. In case, the driver missed or failed to follow the instructions produced by driver monitoring system, the signal of potential emergency situation inside the vehicle cabin is send to the dispatch center to act.

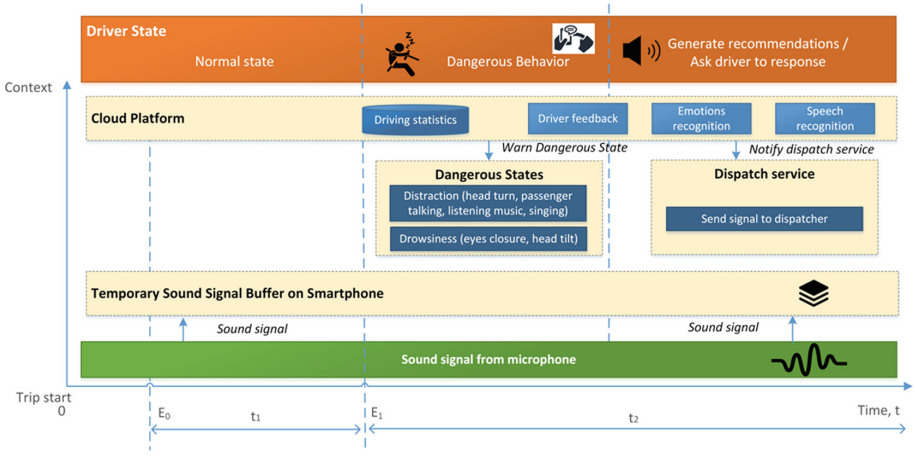


Fig. 3. Scheme of dangerous state determination using audio signal on the timeline.

5 Implementation on Smartphone

The proposed approach for driver behavior monitoring based on audio-visual analysis was tested in Drive Safely mobile application developed for Android platform. Underneath, the application essentially utilizes the front-facing camera for continuously tracking driver facial features (the positions and sizes of head, eyes, mouth) and determining the situations when the driver's eyes and mouth are opened or closed, head is rotated left or right, tilted up or down. Context-based recommendations provided for a driver in a form of a speech-based request to mitigate the emergency situation, require driver to follow it.

To ensure, the driver followed the given instructions, the microphone may be utilized by the mobile application to capture audio signal and to verify the situations whether the driver increased or decreased the music volume of vehicle infotainment system, or he/she talks with passenger. Moreover, to make sure, the driver is concentrated on the road ahead, the smartphone microphone can be used to check the audio driver response asking the driver to say key phrase from the predefined list e.g. "Yes", "Fine", "Great", "Super", etc. In this way, the signal, captured by smartphone microphone, can be analyzed for the presence of audible signals of low or high frequency sound, or the presence of human speech of all vehicle passengers. At the same time, in order to cheer up the driver's attention and reduce the drowsiness state, the mobile application can involve driver in completely hands-free activities by initiating a dialog in a question-answer format (e.g. answering factual information), playing in a last letter word game e.g. Cities, or taking part in a song quiz by guessing the current playing song. The example of driver engagement while drowsiness state is determined by the mobile application is presented in Fig. 4. The demonstration of the mobile application while recognizing drowsiness is shown in Fig. 5.

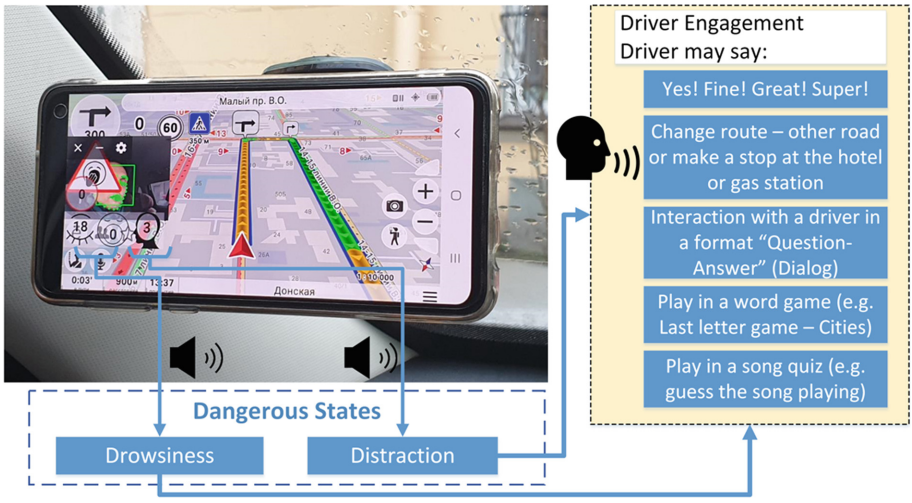


Fig. 4. Driver engagement when drowsiness state is determined by Driver Safely mobile application

Fig. 5. Drowsiness state in driver behavior recognized by Drive Safely mobile application in vehicle cabin

6 Conclusion

In this paper, we proposed the audio-visual approach for dangerous state detection in vehicle cabin based on the data from smartphone sensors. Underneath, this approach differs from existing solutions by combining the use of images frames, containing the driver's face, captured by front-facing camera, and processing audio signal obtained from the microphone of the smartphone to recognize abnormal driving behavior. It should be noted, the presented audio-based analysis is free from calibration effects. The proposed approach was implemented and tested in Drive Safely mobile application on the smartphone [18]. The results of the study we present indicate that our solution may be also successfully applied in rapid development category of advanced driver assistance systems, enhancing dangerous state determination by utilizing microphones to recognize audio cues of abnormal driving behavior. Albeit driver assistance systems already integrated in vehicles, having high-precision sensors, demonstrate high efficiency in determining driving dangerous states under different weather conditions, the mobile applications developed for smartphones are at much lower price, and are increasingly popular among people of any age in many countries, are easy to use in any vehicle. Moreover, to address vehicle specific environment the approach considers most suitable for a driver and permitted by traffic rules hands-free vehicle-driver interaction, allowing drivers to keep eyes on the road and their hands on the wheel. The limitations of the

study are mainly related with poor accuracy and, consequently, decreased performance of speed engine used to recognize text spoken by the driver and to output voice recommendations inside the vehicle cabins that can be extremely noisy. Noisy conditions caused across different driving conditions (e.g. opened windows, speed increase), and maneuver operations may significantly influence the overall system performance. Also, the use of proposed approach, means that the driver voluntarily agrees to follow audio-based recommendations prompted by the system which goal is to help driver to avoid emergency road situation. Thus, our developed approach belongs to zero level vehicle autonomy. As a further improvement of the proposed approach based on audio-visual recognition, combining both audio and video information may largely improve speech recognition over using only the audio information under noisy conditions inside the vehicle cabin. Additionally, the use of other sensors giving quantitate measurements and information about driver's emotions, psychophysiological behavioral state, can provide more valuable feedback for a system to decide whether the driver in dangerous situation or not.

Acknowledgments. Reference model of the driver monitoring system has been developed in scope of Russian Foundation for Basic Research project # 17-29-07073. Audiovisual approach for dangerous state determination is supported by the Russian Foundation for Basic Research project # 19-29-09081. Implementation has been done in scope of Russian State Research # 0073-2019-0005.

References

1. 2018 road safety statistics: what is behind the figures? https://ec.europa.eu/commission/presscorner/detail/en/MEMO_19_1990. Accessed 14 Dec 2019
2. The EU Wants Cars To Have Speed Limiters and More by 2022. https://www.roadandtrack.com/new-cars/car-technology/a26960542/the-eu-wants-cars-to-have-speed-limiters-and-more-by-2022/. Accessed 14 Dec 2019
3. Volvo Vision 2020. https://www.unece.org/fileadmin/DAM/trans/roadsafe/unda/Sweden_Volvo_Vision_2020.pdf. Accessed 14 Dec 2019
4. Kashevnik, A., Lashkov I.: Decision support system for drivers passengers: smartphone-based reference model and evaluation. In: Conference of Open Innovation Association, FRUCT, pp. 166–171. IEEE Computer Society (2018)
5. Kashevnik, A., Lashkov, I., Gurtov, A.: Methodology and mobile application for driver behavior analysis and accident prevention. IEEE Trans. Intell. Transp. Syst. **21**(6), 1–10 (2019)
6. Harrison, Y., Horne, J.A.: Sleep deprivation affects speech. J. Sleep Res. Sleep Med. **20**(10), 871–877 (1997)
7. Krajewski, J., Batliner, A., Golz, M.: Acoustic sleepiness detection: framework and validation of a speech-adapted pattern recognition approach. Behav. Res. Methods **41**, 795–804 (2009)
8. Batliner, A., Steidl, S., Nöth, E.: Releasing a thoroughly annotated and processed spontaneous emotional database: the FAU Aibo Emotion Corpus (2008)
9. Martin, V. P., Rouas, J.-L., Thivel, P., Franchi J.-A. M., Philip, P.: Towards automatic sleepiness measurement through speech, pp. 1–10 (2019)
10. Shahid, A., Wilkinson, K., Marcu, S., Shapiro, C.M.: Karolinska sleepiness scale (KSS). Psychology (2011)

11. Fraile, R., Godinol lorente, J.: Cepstral peak prominence: a comprehensive analysis. Biomed. Sig. Process. Control. **14**(1), 42–54 (2014)
12. Fernandes, J., Teixeira, F., Guedes, V., Junior, A., Teixeira, J.: Harmonic to noise ratio measurement - selection of window and length. Proc. Comput. Sci. **138**, 280–285 (2018)
13. Prokhorov, D., Kalik S., Varri C.: Toyota motor corp. system and method for reducing boredom while driving. US7982620B2, United States Patent and Trademark Office, 19 June 2011
14. Martin, V.P., Rouas, J., Thivel, P., Krajewski, J.: Sleepiness detection on read speech using simple features. In: 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Timisoara, Romania, pp. 1–7 (2019)
15. Kamaruddin, N., Wahab, A.: Heterogeneous driver behavior state recognition using speech signal. In: Proceedings of the 10th WSEAS International Conference on Power Systems and Systems Science, pp. 207–212 (2011)
16. Angkititrakul, P., Kwak, D., Choi, S., Kim, J., PhucPhan, A., Sathyanarayana, A., Hansen, J. H. L.: Getting start with UTDrive: driver-behavior modeling and assessment of distraction for in-vehicle speech systems. In: INTERSPEECH-2007 – 8th Annual Conference of the International Speech Communication Association, Belgium, pp. 1334–1337 (2007)
17. Sommer, D., Golz, M.: Evaluation of PERCLOS based current fatigue monitoring technologies. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, pp. 4456–4459 (2010)
18. Google Play – Drive Safely. https://play.google.com/store/apps/details?id=ru.igla.drivesafely. Accessed 14 Dec 2019

Author Index

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1250, pp. 801–803, 2021.
https://doi.org/10.1007/978-3-030-55180-3