PP-RAI'2022

Proceedings of the 3rd Polish Conference on Artificial Intelligence
April 25-27, 2022, Gdynia, Poland

Publishing House of Gdynia Maritime University
Gdynia 2022

These proceedings were supported by the statutory funds of the Department of Information Systems, Management and Quality Science, Gdynia Maritime University and Gdynia Maritime University.

Editorial layout and cover design by Piotr Jędrzejowicz, Ireneusz Czarnowski, Aleksander Skakovski, Marcin Forkiewicz, Marta Szarmach and Paweł Wolski.

ISBN: 978-83-7421-401-8

Publisher:

Gdynia Maritime University
Publishing House of Gdynia Maritime University
ul. Morska 81-87, 81-225 Gdynia, Poland

Preface

This book contains the proceedings of the 3rd Polish Conference on Artificial Intelligence (PP-RAI'2022). The conference was held in Gdynia, Poland, on April 25-27, 2022. The conference was held under the scientific supervision of the Polish Alliance for the Development of Artificial Intelligence.

Polish Alliance for the Development of Artificial Intelligence (pol.: *Polskie Porozumienie na Rzecz Rozwoju Sztucznej Inteligencji* - PP-RAI) was established in 2018 based on the agreement and declaration of mutual cooperation to achieve the goal of facilitating and supporting the AI development in Poland. The following scientific organizations were the parties of this agreement:

- PSSI - Polskie Stowarzyszenie Sztucznej Inteligencji,
- PTSN - Polskie Towarzystwo Sieci Neuronowych,
- PL SIGML - Polska Grupa Systemów Uczących się,
- IEEE SMC · Polski Oddział IEEE SMC (Polish Chapter of the IEEE Systems, Man, and Cybernetics Society),
- IEEE CIS · Polski Oddział IEEE Computational Intelligence Society.

The PP-RAI provides opportunities for the presentation of new and original research results in different areas of artificial intelligence. The PP-RAI also provides a national forum for sharing, exchanging, presenting, and discussing ideas relevant to the development of the AI methods and tools. The Alliance encourages the growth of social interest in AI and suggests strategic directions of research and applications, encouraging interdisciplinary involvement and cooperation.

This year's conference included 3 plenary lectures offered by internationally recognized experts, 4 discussion panels on hot topics relevant to AI development, and 110 theoretical and application papers presented during 12 scientific sessions. The proceedings contain only papers reviewed and selected by the Program Committee and the Reviewer Board from all presented at the scientific sessions.

We would like to thank all the authors for their effort and paper preparation and presentations. We also gratefully acknowledge the hard work of the track chairs, as well as, program committee members and the additional reviewers for their input and involvement. Their contribution has made an important impact on the PP-RAI'2022 conference program.

Piotr Jędrzejowicz
Ireneusz Czarnowski

April, 2022

Organization

Honorary Patronage

Mieczysław Struk	Marshal of the Pomorskie Voivodeship, Poland
Wojciech Szczurek	Mayor of Gdynia, Poland

General Chair

Adam Weintrit	Rector of Gdynia Maritime University, Poland

Steering Committee

Włodzisław Duch	Nicolaus Copernicus University in Toruń, Poland
Krzysztof Dembczyński	Poznan University of Technology, Poland
Janusz Kacprzyk	Systems Research Institute Polish Academy of Sciences, Warsaw, Poland
Jacek Koronacki	Institute of Computer Science Polish Academy of Sciences, Warsaw, Poland
Jacek Mańdziuk	Warsaw University of Technology, Poland
Grzegorz J. Nalepa	Jagiellonian University in Kraków, Poland
Leszek Rutkowski	Częstochowa Univerity of Technology, Poland
Jerzy Stefanowski	Poznan University of Technology, Poland
Halina Kwaśnicka	Wrocław University of Science and Technology, Poland
Ireneusz Czarnowski	Gdynia Maritime University, Poland
Dominik Ślęzak	University of Warsaw, Poland
Michał Woźniak	Wrocław University of Science and Technology, Poland

Program Chair

Piotr Jędrzejowicz	Gdynia Maritime University, Poland
Ireneusz Czarnowski	Gdynia Maritime University, Poland

Publicity Chair

Paweł Zyblewski	Wrocław University of Science and Technology, Poland
Izabela Wierzbowska	Gdynia Maritime University, Poland

Organizing Committee (Local Team - Gdynia Maritime University)

Ireneusz Czarnowski	
Dariusz Barbucha	
Łukasz Wyszyński	Fundacja PSC
Ewa Ratajczak-Ropel	
Natalia Mańkowska	

Paweł Szyman
Ireneusz Meyer
Marta Szarmach
Krzysztof Sadowski
Iza Wierzbowska
Marcin Forkiewicz
Aleksander Skakovski
Marta Stempniewicz
Paweł Wolski

Track Chairs

Computer Vision

Maciej Piasecki	Wrocław University of Science and Technology, Poland

Data Mining and Machine Learning

Jerzy Stefanowski	Poznan University of Technology, Poland
Michał Woźniak	Wrocław University of Science and Technology, Poland

Knowledge Engineering

Agnieszka Lawrynowicz	Poznań University of Technology, Poland
Grzegorz Nalepa	Jagiellonian University in Kraków, Poland
Dariusz Król	Wrocław University of Science and Technology, Poland

Neuroinformatics

Włodzisław Duch	Nicolaus Copernicus University in Toruń, Poland
Juliusz Szymański	Gdańsk University of Technology, Poland

Natural Language Processing, Automatic Speech Recognition and Conversational AI

Maciej Piasecki	Wrocław University of Science and Technology, Poland

Problem Solving and Optimization

Ewa Bednarczuk	Warsaw University of Technology and Systems Research Institute Polish Academy of Sciences, Poland

Robotics and Autonomous Systems

Piotr Skrzypczyński	Poznan University of Technology, Poland

Uncertainty in Artificial Intelligence

Dominik Ślęzak	University of Warsaw, Poland

Young.AI - Session for young researchers

Agnieszka Lazarowska	Gdynia Maritime University, Poland

Program Committee and Reviewers

Piotr Artiemjew
Michal Baczyński
Janusz Bedkowski
Ewa Bednarczuk
Dominik Belter
Aleksander Byrski
Leszek Chmielewski
Ireneusz Czarnowski
Michał Czubenko
Krzysztof Dembczyński
Włodzisław Duch
Soma Dutta
Paweł Forczmański
Dariusz Frejlichowski
Piotr Gawron
Agata Giełczyk
Andrzej Janusz
Joanna Jaworek-Korjakowska
Paweł Jemioło
Piotr Jędrzejowicz
Janusz Kacprzyk
Katarzyna Kaczmarek-Majer
Aleksandra Knapińska
Tomasz Kornuta
Jacek Koronacki
Anna Korzyńska
Marek Kraft
Dariusz Król
Paweł Ksieniewicz
Marcin Kurdziel
Halina Kwasnicka
Bogdan Kwolek
Agnieszka Lawrynowicz
Agnieszka Lazarowska
Anna Lewandowska
Jacek Mańdziuk
Andrzej Materka
Magdalena Mazur-Milecka
Wojciech Moczulski
Mostefa Mohamed-Seghir
Grzegorz Nalepa
Mariusz Nieniewski
Michał Nowicki
Boguslaw Obara
Krzysztof Okarma
Mete Ozay
Barbara Pekala
Maciej Piasecki
Ewa Ratajczak-Ropel
Przemyslaw Rokita
Jacek Ruminski
Leszek Rutkowski
Monika Rybczak
Rafał Scherer
Andrzej Skowron
Piotr Skrzypczyński
Łukasz Sosnowski
Jerzy Stafanowski
Katarzyna Stapor
Maciej Stefańczyk
Michał, Strzelecki
Marcin Szeląg
Julian Szymanski
Dominik Ślęzak
Andrzej Śluzek
Tomasz Trzcinski
Krzysztof Walas
Jarosław Wąs
Izabela Wierzbowska
Anna Witkowska
Konrad Wojciechowski
Marcin Woźniak
Michal Woźniak

Table of Contents

A Performance Improvement of Deep Learning Based Binarization of Degraded Document Images with the Use of the Voting Approach

Hubert Michalak[0000−0003−4888−4303] and Krzysztof Okarma[0000−0002−6721−3241]

West Pomeranian University of Technology in Szczecin
Faculty of Electrical Engineering
Department of Signal Processing and Multimedia Engineering
26 Kwietnia 10, 71-126 Szczecin, Poland

Abstract. In recent years a great progress and interest in the use of deep learning in computer vision applications may be observed, often leading to encouraging results. This also concerns image binarization methods, particularly for degraded document images where adaptive thresholding should be applied. Nevertheless, the use of neural networks, trained using relatively small datasets, may suffer from potential overfitting and the application of such trained deep networks for some other datasets does not always lead to satisfactory results. To increase the universality of such methods, the approach based on pixel voting has been proposed. In this approach, multiple methods, including those based on deep learning, are applied in parallel, and the final result depends on the majority of binarization results obtained using these methods at the pixel level. As verified for state-of-the-art datasets, the proposed approach leads to significant performance improvement in comparison to the other methods.

Keywords: Image binarization · Document images · Deep learning.

1 Introduction

Binarization of degraded document images, as well as natural images captured in uncontrolled lighting conditions, is still one of the active field of research in image processing and analysis. Such image preprocessing methods are useful not only for further text recognition in document images or the analysis of 2D binary codes but also for navigation of mobile robots, e.g., line followers, particularly in unstable lighting conditions. Therefore, highly degraded document images, particularly unevenly illuminated, are commonly accepted as benchmarks for newly proposed image binarization methods. Starting from well-known DIBCO datasets, recently some even more demanding datasets have been delivered, e.g. Bickley Diary, Nabuco and LiveMemory datasets or MonkCuper database. Some of them are available on the DIB website[1] hosted by Brazilian Universidade Federal de Pernambuco (UFPE).

[1] https://dib.cin.ufpe.br

Although for well-illuminated high quality images, Niblack-based adaptive thresholding methods, such as Sauvola, Wolf, Feng and NICK [1], or even global binarization methods, such as the most popular Otsu, might be enough, more demanding images require the development of more sophisticated solutions. One of the possible approaches is the use of deep neural networks, although such methods require time-consuming training and are not necessarily fast enough, as reported e.g. at the ICDAR 2021 Competition on Time-Quality Document Image Binarization [2]. Additionally, an important problem of such approaches may be their overfitting observed when the datasets used for training differ from testing images, considering image size, distortions, illumination, etc.

To demonstrate this issue, two deep-learning (DL) based methods have been used in the paper: the first developed by Sami Liedes[2] who trained the network using mainly DIBCO and Persian datasets, and the second, known as RObust document image BINarization tool (ROBIN), developed by Mikhail Masyagin[3]. Both tools have been written in Python in combination with some other open source projects such as OpenCV, Keras, Tensorflow or Augmentor, as reported in their documentation. An improvement of their performance, partially solving the overfitting problem, is possible using the pixel voting approach presented below.

2 Proposed Approach

Considering the progress in the development of modern processing units, there are wide possibilities of parallel processing of images, also using multiple methods independently. Hence, regardless of the relatively long computation time, observed for DL-based methods, their combination with some other approaches might be beneficial, potentially improving the obtained results, without significant increase of the total processing time. One of such approaches, considered in the paper, is the application of the pixel voting, successfully applied previously for some other image thresholding methods used for image preprocessing before the alphanumerical character recognition [5].

The main assumption of the pixel voting is the independent parallel binarization of an input image using a number of N selected algorithms where each pixel in the resulting image may be expressed as 0 or 1. Treating each ot these values as votes, the "winning" value may be selected as the final result for the considered pixel. Actually, this method may also be implemented simply as the median of the obtained binary values using each method at the pixel level. Nevertheless, as presented in the paper [5], satisfactory results may be obtained by the application of some algorithms based on various assumptions and the combination of similar algorithms, e.g. only Niblack-inspired adaptive methods, usually leads to worse performance, particularly for non-uniformly illuminated document images. Hence, the combination of the DL-based methods with some others seems to be an appropriate assumption for further investigation.

[2] https://github.com/sliedes/binarize
[3] https://github.com/masyagin1998/robin

3 Experimental Results

The verification of the proposed use of the pixel voting has been made utilizing commonly used datasets, including DIBCO 2009–2019, Bickley Diary, Persian, Nabuco (part 1 with 15 images and part 2 – 20 images), LiveMemory (20 images) and Monk Cuper Set (25 images). Since the verifications have been made calculating some typical metrics [6] such as F-Measure, Accuracy, Distance Reciprocal Distortion (DRD) and Misclassification Penalty Metric (MPM), only the images with known ground-truth (GT) binary images may be used for this purpose. To illustrate the obtained results in a relatively compact representative form, only the obtained F-Measure (FM) values are presented in Table 1.

The previously proposed methods utilizing the stack of regions [3] are marked in Table 1 as 1L, 8L and 16L, according to the number of layers used in the calculations. The five sets of methods used in the pixel voting, the results for which are shown in Table 1, are the combinations of ROBIN with: 1L (single-layered) and JUCS (set 1), 16L and JUCS (set 2), Niblack and JUCS (set 3), 8L and Sauvola (set 4), and JUCS with Sauvola (set 5). The three best results for each dataset are marked by boldface fonts. The last row contains the results obtained for all datasets except DIBCO 2009–2018 (used for training of the DL-based methods). It may be easily noted that the best results obtained for DIBCO 2009–2018 datasets used for training are the effect of the overfitting.

Application of the deep-learning based methods for the other datasets, containing previously unseen data, leads to significantly worse results. Nevertheless, the use of the pixel voting, also in combination with some other previously proposed algorithms, makes it possible to improve the results of binarization.

Table 1. F-Measure values obtained for various datasets using some classical methods, the use of stack of regions, the JUCS method [4], deep learning methods and the proposed pixel voting.

Method	classical methods				stack of regions			JUCS	DL		Pixel Voting				
/ Dataset	Otsu	Nibl	Sauv	Brad	1L	8L	16L	[4]	ROBIN	Liedes	set 1	set 2	set 3	set 4	set 5
DIBCO 2009	78.6	76.8	78.7	77.0	77.1	81.3	81.4	84.8	**93.1**	71.8	87.1	87.0	86.7	**88.7**	**89.4**
DIBCO 2010	85.4	78.0	81.1	82.8	79.2	81.1	81.2	82.4	**94.2**	56.4	86.5	86.1	**87.1**	**86.7**	86.5
DIBCO 2011	82.1	68.9	78.7	74.9	72.6	75.4	75.5	81.0	**91.5**	65.0	83.8	83.4	83.3	**85.6**	**86.6**
DIBCO 2012	75.1	77.2	81.1	82.3	79.3	82.6	82.7	85.4	**94.7**	65.6	88.5	88.2	**89.0**	88.5	**88.9**
DIBCO 2013	80.0	72.7	78.8	77.5	76.1	78.9	78.9	82.7	**94.7**	71.4	86.3	85.9	86.3	**86.4**	**86.6**
DIBCO 2014	91.6	84.9	90.3	88.9	84.5	86.4	86.4	89.5	**96.0**	66.6	91.8	91.5	91.6	**93.7**	**94.5**
DIBCO 2016	86.6	74.2	80.1	76.0	76.3	81.1	81.2	86.1	**90.6**	68.3	87.5	87.5	87.2	**87.6**	**88.4**
DIBCO 2017	77.7	75.0	77.9	76.6	75.6	79.7	79.7	82.7	**92.2**	56.7	85.7	85.5	85.2	**86.3**	**86.9**
DIBCO 2018	51.5	67.6	54.6	61.0	68.0	70.3	70.4	72.5	**88.7**	57.6	**79.7**	78.5	**79.5**	76.4	75.7
DIBCO 2019A	**72.3**	54.4	48.1	54.3	58.4	60.8	61.0	71.1	46.9	26.6	**71.3**	**71.4**	70.8	65.4	70.5
DIBCO 2019B	23.3	54.2	44.4	42.1	52.1	54.4	54.4	58.2	43.3	34.2	**61.0**	**60.4**	**60.3**	59.0	59.6
Bickley Diary	58.8	83.8	72.4	70.6	78.9	84.4	84.5	83.6	73.1	42.8	**86.6**	**87.9**	**88.8**	86.5	83.2
Persian	82.1	78.2	**86.8**	81.2	79.2	83.0	83.1	85.8	85.5	72.6	86.6	86.7	86.0	**88.1**	**88.4**
Nabuco part 1	86.3	77.1	75.3	74.8	78.7	82.1	82.2	**87.1**	86.1	70.9	87.0	**87.2**	**87.4**	83.8	85.1
Nabuco part 2	**94.0**	82.8	83.3	81.0	84.6	88.6	88.6	**93.8**	90.2	71.9	92.9	**93.2**	93.1	90.0	91.3
Livememory	89.6	90.2	89.0	88.7	91.4	93.0	93.0	**94.8**	82.4	42.4	**93.0**	**93.9**	93.0	90.0	90.7
Monk Cuper Set	69.4	68.9	73.4	70.3	70.1	73.1	73.1	77.2	**84.8**	73.0	78.9	78.6	78.4	**79.0**	**79.8**
All	71.9	77.2	75.0	73.7	76.7	80.5	80.6	83.0	81.9	56.4	**85.2**	**85.5**	**85.8**	84.7	84.3
Verification	68.1	78.5	73.3	71.6	76.9	81.1	81.2	83.0	75.8	52.1	**84.7**	**85.3**	**85.6**	83.6	82.7

An illustration of the results obtained for a sample image from DIBCO2019 dataset, presenting the advantages of the pixel voting, is shown in Fig. 1.

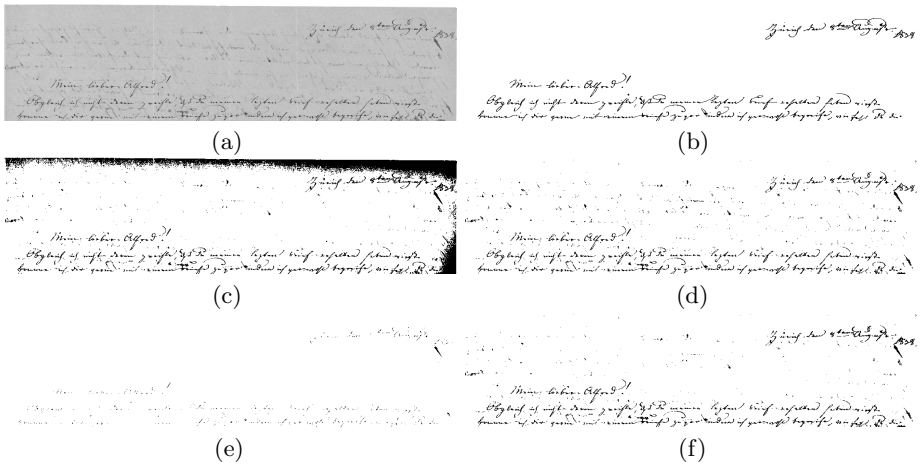


Fig. 1. Results obtained for a sample image from DIBCO2019 dataset track A, filename *02.bmp*: (a) input image, (b) ground truth, (c) Sauvola $FM = 36.12$, (d) JUCS $FM = 78.12$, (e) DL ROBIN $FM = 64.65$, (f) pixel voting – set 5 $FM = 81.72$.

References

1. Khurshid, K., Siddiqi, I., Faure, C., Vincent, N.: Comparison of Niblack inspired binarization methods for ancient documents. In: Document Recognition and Retrieval XVI. vol. 7247, pp. 7247–7247–9. SPIE (2009). https://doi.org/10.1117/12.805827
2. Lins, R.D., Bernardino, R.B., Smith, E.B., Kavallieratou, E.: ICDAR 2021 competition on time-quality document image binarization. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 708–722. Springer International Publishing, Cham (2021)
3. Michalak, H., Okarma, K.: Adaptive image binarization based on multi-layered stack of regions. In: Vento, M., Percannella, G. (eds.) Computer Analysis of Images and Patterns. LNCS, vol. 11679, pp. 281–293. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-29891-3_25
4. Michalak, H., Okarma, K.: Fast binarization of unevenly illuminated document images based on background estimation for optical character recognition purposes. Journal of Universal Computer Science **25**(6), 627–646 (2019). https://doi.org/10.3217/jucs-025-06-0627
5. Michalak, H., Okarma, K.: Robust combined binarization method of non-uniformly illuminated document images for alphanumerical character recognition. Sensors **20**(10), 2914 (may 2020). https://doi.org/10.3390/s20102914
6. Ntirogiannis, K., Gatos, B., Pratikakis, I.: Performance evaluation methodology for historical document image binarization. IEEE Transactions on Image Processing **22**(2), 595–609 (2013). https://doi.org/10.1109/TIP.2012.2219550

Convolutional Sparse Coding of Images through Characteristic Points Extraction

Damian Salata, Arkadiusz Tomczyk[0000−0001−9840−6209], and Piotr S. Szczepaniak[0000−0002−9973−0673]

Institute of Information Technology
Lodz University of Technology
al. Politechniki 8, 93-590 Lodz, Poland
arkadiusz.tomczyk@p.lodz.pl

Abstract. Sparse coding refers to methods that allow to reveal, in an unsupervised way, internal structure hidden in the data. When applied to images, it enables their automatic decomposition into a set of semantic elements. There are several classic algorithms that allow to address sparse coding problem. In this work we show that one can adapt for that purpose a convolutional neural network, which originally was used to solve completely different task (characteristic points extraction). Our contribution includes both network architecture and a method of its training.

Keywords: Sparse coding · Characteristic points · Convolutional neural networks · Image analysis.

1 Introduction

Sparse coding and characteristic points (landmarks) extraction are two, usually not connected, groups of techniques which find multiple applications in image analysis. The first ones are an unsupervised learning algorithms able to discover a set of basis functions that capture higher-level features in the data. When image data are taken into account, images are automatically decomposed into a set basic elements, which allows to reveal their internal structure and consequently leads to their alternative, reduced representation. It finds its application in: image classification, noise removal, compression, image synthesis, etc. Moreover making sparse codes is considered as a plausible model of the visual cortex ([4]). The second, are methods for detecting and describing local features of the images. Well known classic techniques used for that purpose are: SIFT, SURF, ORB or BRISK ([6, 1, 7, 5]). Characteristic points were initially based on corners, which are relatively easy to find. In recent years, however, development of convolutional neural networks CNN ([3]) made it possible to use them for this task as well. This allowed to extract characteristic points, which did not necessarily have to be corners. One of the first algorithms based on CNNs, which simultaneously determined characteristic points and their descriptors, was SuperPoint ([2]). Characteristic points are used for: object detection, 3D reconstruction,

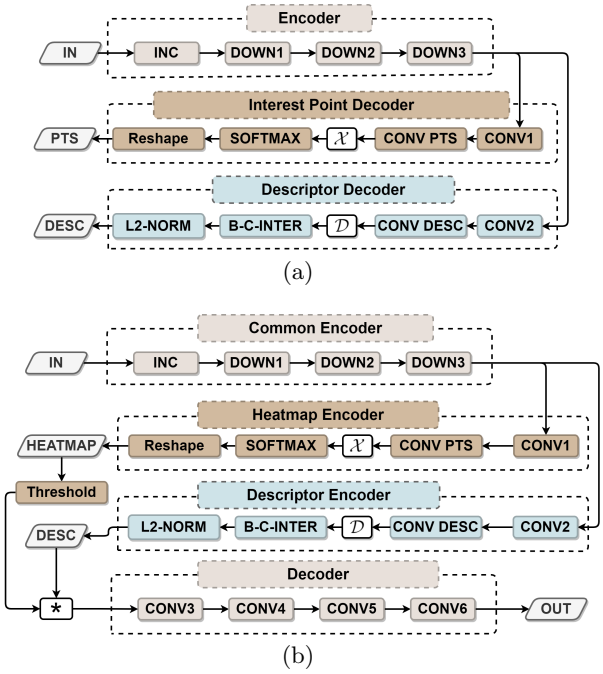


Fig. 1. The difference between original SuperPoint architecture (a) and its modified version used in this work (b). The main difference is abandonment of characteristic point extraction PTS and combination of thresholded HEATMAP with DESC to create the input of the decoder (they contain sparse code of the input image).

panorama stitching by combining different images, object tracking, video stabilization, etc.

In this paper we propose a novel method for image sparse coding which bases on characteristic points extraction algorithm. The original contribution of this paper covers: encoder basing on landmark detector, 2D sparse codes with non-linear decoder, loss functions and two-stage training procedure.

2 Method

In our approach we use a modified SuperPoint[1] convolutional characteristic points extractor, which architecture is presented in Figure 1a, as an encoder part of autoencodoer shown in Figure 1b. This architecture was chosen since its working principle enables an easy imposition of sparsity constraints. The original method for a given input image of size $W \times H$ generates two outputs $\mathcal{X}$

[1] Due to limited number of pages, only necessary details of this method are presented in this work - we encourage reader to get familiar with original work [2].

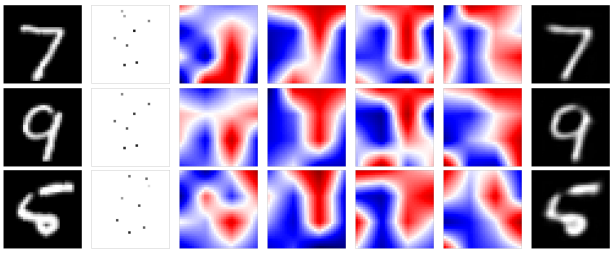


Fig. 2. Sample reconstructions of selected MNIST test images, from left there are: input image, thresholded HEATMAP, all channels of DESC map and network output.

and $\mathcal{D}$ of reduced size $W/8 \times H/8$, which after further processing, lead to positions of landmarks PTS and d-dimensional descriptor map DESC (after bicubic interpolation size of DESC is the same as the size of input image).

Our modifications were motivated by both the need of adding convolutional decoder and the necessity of proper loss function $\mathcal{L}$ definition. We do not extract PTS from $\mathcal{X}$ but generate HEATMAP containing probabilities of landmarks existance separately in non-overlapping 8×8 blocks. In order to prevent decoder learning from insignificant codes we threshold the HEATMAP to remove small values (values below 0.1 are set to be 0). Finally, since both thresholded HEATMAP and DESC are of the same size $W \times H$, we multiply them elementwise to create the input of the decoder. To define sparsity constraints we use properties of $\mathcal{X}$ ([2]). It contains vectors with 65 elements. First 64 values are later interpreted as probabilities of landmarks in HEATMAP blocks (after reshaping), whereas the last one indicates the probability of landmark absence in the considered block. Since, within every block, the sum of all those probabilities is equal to 1 (thanks to the softmax operation) to force minimum number of landmarks (sparsity) we can expect a value close to 1 at the last position of $\mathcal{X}$ vectors. To express less restrictive requirement of having only one or none significant landmark, we can anticipate that maximum value of vectors in $\mathcal{X}$ should be close to 1. Further, we denote constraints taking into account the above observations as $\mathcal{L}^{hard}$ and $\mathcal{L}^{soft}$, respectively.

Autoencodoer is trained to reconstruct its input on the output. That is why the main component of the loss function is a mean squared error $\mathcal{L}^{mse}$ between network output and input. The preliminary experiments have shown that direct usage of hard sparsity constraint does not allow finding the desired optimum. That is why, we propose a two-stage strategy where at the begining network is trained using $\mathcal{L} = \mathcal{L}^{mse} + \lambda \cdot \mathcal{L}^{soft}$, and later the $\mathcal{L} = \mathcal{L}^{mse} + \lambda \cdot \mathcal{L}^{hard}$ is applied.

3 Results

To show the properties of the proposed approach we have trained autoencoder using MNIST dataset with hand-written digits. In the conducted experiments the descriptor size d was equal to 4 and $\lambda = 0.1$. Sample reconstruction results

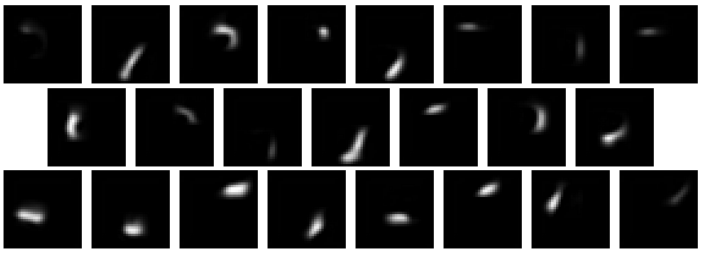


Fig. 3. Visual interpretation of discovered codes for images presented in Figure 2 (landmark distribution as well as their number are different for different images).

together with codes (thresholded HEATMAP) and descriptor maps (DESC) are presented in Figure 2. To show how images are decomposed or, in other words, to understand what is a visual interpretation of automatically discovered codes, we have used a decoder to process every landmark separately (Figure 3).

4 Summary

In this work we have shown how modified architecture of SuperPoint network, originally used for characteristic point extraction, can be applied for sparse coding of the given class of images. The proposed method was tested using relatively simple MNIST dataset. Further experiments will focus on more complex datasets and on the usage of generated, reduced image representations for image analysis related tasks such as: classification, segmentation, etc. In particular we are interested in application of graph convolutional neural networks able to directly operate on alternative representations of image content.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. vol. 110, pp. 404–417 (01 2006)
2. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. CoRR **abs/1712.07629** (2017)
3. Lecun, Y., Bengio, Y.: Convolutional Networks for Images, Speech and Time Series, pp. 255–258. The MIT Press (1995)
4. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems. vol. 19. MIT Press (2006)
5. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision. pp. 2548–2555 (2011). https://doi.org/10.1109/ICCV.2011.6126542
6. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1150–1157 vol.2 (1999). https://doi.org/10.1109/ICCV.1999.790410
7. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. pp. 2564–2571 (11 2011). https://doi.org/10.1109/ICCV.2011.6126544

Counting Dots: On Learning Numerical Concepts from Visual Data

Andrzej Śluzek[0000−0003−4148−2600]

Institute of Information Technology,
Warsaw University of Life Sciences-SGGW, Poland
andrzej_sluzek@sggw.edu.pl

Abstract. The paper reports selected ideas and preliminary results on NN-based learning simple numerical properties from visual data. Specifically, we use datasets of small images with single-pixel dots (*one* to *ten* dots in each image) to visually estimate integers (ranging from 1 to 10), to distinguish between *even* and *odd* numbers, and to identify some other numerical properties. Small fully-connected NN's and convolutional NN's are used as learning architectures. The obtained results are inconclusive. First, it seems that FCNN architectures are superior to CNN's, which are apparently hardly able to learn the abstractions of numbers. Surprisingly, such conclusions contradict results presented in a recently published paper on similar topics. Secondly, the concepts of *even* and *odd* numbers cannot be learned directly from visual examples, even by FCNN's. We preliminarily hypothesize that basic numerical properties can or cannot be learned by simple NN architectures depending on whether the property divides the considered set of integers into connected subsets (e.g. 1, 2 and larger numbers) or disjointed ones (e.g. even *versus* odd numbers). Nevertheless, the obtained results are still considered preliminary, and they require further theoretical analysis and experimental verification.

Keywords: Numerical Abstractions · Learning · Neural Networks · Visual Data.

1 Introduction and Background

Counting is one of the first abstract concepts developed by children, e.g. [8]. Intuitively, and based on various publications (e.g. [4, 3]) this concept develops from sensory experiences and physical embodiments, with visual inspection playing the vital role.

Machine learning community has been interested in such issues for a long time. Counting and understanding numerical concepts from visual data is an interesting and challenging topics for AI algorithms. Initially, the majority of papers were focused on automatic object counting rather than on 'understanding' the abstractions of numbers, with significant efforts on *counting-by-localizing* sub-tasks, e.g. [5, 7]. This was, obviously, more application-oriented and various

sources (e.g. [2, 6]) indicate that true understanding of numbers may not be needed to automatically perform counting tasks.

This paper follows another approach (presented in fewer papers) where visual complexity of training/validation data is reduced, and the focus is on learning the concept of numbers (including abilities to estimate the numbers of objects). In particular, we follow and expand the approaches from [1], where two models are proposed for learning counting (up to 10) in small black images containing a number of white non-overlapping rectangles.

The authors used a standard feed-forward CNN (of very low complexity) and a deep believe network(DBN) model, reporting superior performances of the former, and providing details of the implemented architecture. Compared to that paper, we investigate similar problems. However, other visual datasets and alternative architectures are used. More diversified numerical concepts are considered, and (notably) we obtain results which partially contradict [1].

2 Methodology and Objectives

2.1 Datasets

For the conducted experiments, we generated a simple dataset of 20,000 images. As shown in Fig. 1, image resolution is 10×10 with a number of isolated single-pixel dots over backgrounds of opposite intensities. The images are not binary to better represent the real-world visual conditions (and to minimize effects of overfitting on binary data).

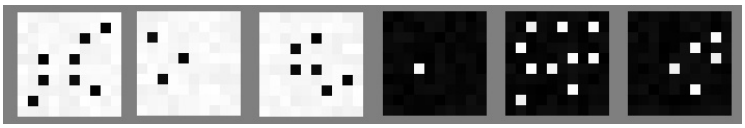


Fig. 1. Exemplary dataset images.

2.2 Tasks

Several learning tasks have been attempted by training selected NN architectures of the dataset. Actually, the dataset is divided into two equal halves, and training/validation is always performed on an arbitrarily selected half (the other one used for extensive testing). The tasks (which are learned separately and independently from the provided visual dataset) are as follows:

a. Learning integers from 1 to 10.
b. Learning even and odd numbers (from the above range).
c. Learning the numbers 1, 2 and larger.

The acquired abilities would be used for estimating the number (or number category) of dots in test images.

2.3 NN Architectures

Following the results reported in [1], we initially tested small CNN's proposed there (with minor changes reflecting differences between the datasets). Eventually, even simpler CNN architectures were used, with much better results.

Nevertheless, because of unsatisfactory performances of CNN's (details in Section 3), fully connected NN's (with correspondingly low complexity) were used as the ultimate choice. For either CNN's and FCNN's, various variants were attempted. Table 1 provides examples of top-performance architectures in each category.

Table 1. Examples of tested architectures (CNN based on [1], our CNN and FCNN).

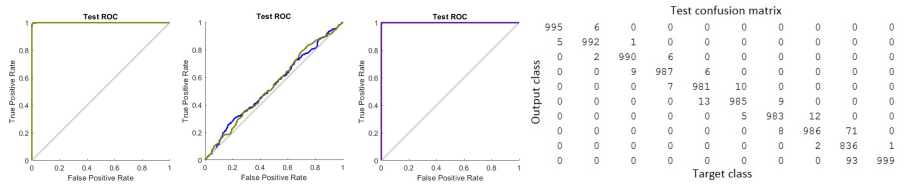
(A) CNN (based on [1])			(B) CNN (our)			(C) FCNN		
Layer	Parameters	Activation	Layer	Parameters	Activation	Layer	Parameters	Activation
input	10×10		input	10×10		input	100	
conv.	3×3, str=1	relu	conv.	2×2, str=2	relu	fc	10	tanh
maxpool	2×2, str=2		maxpool	1×1, str=1		fc	5	tanh
conv.	1×1, str=1	relu	fc	100	tanh	fc	5	tanh
fc	90	tanh	fc	10 (or 2)	softmax	fc	10 (or 2)	softmax
fc	10 (or 2, 3)	softmax	output	10 (or 2, 3)		output	10 (or 2, 3)	
output	10 (or 2)							

3 Experimantal Results

Training of all architectures was performed on 70% of the selected half of the dataset (the remaining 30% used for validation), while the other half of the datatest was used for testing.

Unfortunately, neither CNN architectures similar to Table 1(A) nor CNN architectures similar to Table 1(B) show satisfactory performances for Task(a), i.e. counting from 1 to 10. Accuracy of the former ones is comparable to random choice (10% on both training and testing data). The latter architectures perform better. On training data, the obtained accuracy can exceed 90%, but for validation and testing data we could never get accuracy above 62%. Thus, the experimental results (primarily in a form of ROC curves) are displayed only for FCNN architecture (using the exemplary architecture from Table 1(C)).

As shown in Figs 2(a-d), FCNN almost perfectly handles Task(a) and, in particular, Task(c) (see Section 2.2) but fails to learn the concepts of even and odd numbers (Task(b)). For Task(a), accuracy for test data is 97.34%, while for Task(c) it reaches 99.88%.



(a) (b) (c) (d)

Fig. 2. ROC plots for recognition of numbers: (a) from 1 to 10, (b) *odd* or *even* numbers, (c) numbers 1, 2 or larger. In (d), the confusion matrix (test data) for Task(a) is given.

The results, in spite of very low complexity of applied learning mechanisms, very well correspond to the actual numerical skills of small children (e.g. [8]). Counting *one*, *two*, *many* is the easiest task, counting to 10 comes later (and is more error-prone, with most errors for larger numbers), while understanding the concept of even and odd numbers is seldom achieved before the primary school.

Nevertheless, the obtained results are considered preliminary, and they require further theoretical analysis and experimental verification.

References

1. Creatore, C., Sabathiel, S., Solstad, T.: Learning exact enumeration and approximate estimation in deep neural network models. Cognition **215**, 104815 (2021). https://doi.org/10.1016/j.cognition.2021.104815
2. Fang, M., Zhou, Z., Chen, S., McClelland, J.L.: Can a recurrent neural network learn to count things? Cognitive Science (2018)
3. Fischer, M.H., Shaki, S.: Number concepts: abstract and embodied. Phil. Trans. Royal Society B **373**(1752), 20170125 (2018). https://doi.org/10.1098/rstb.2017.0125
4. Kersey, A.J., Cantlon, J.F.: Primitive concepts of number and the developing human brain. Language Learning and Development **13**(2), 191–214 (2017). https://doi.org/10.1080/15475441.2016.1264878
5. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems - Vol.1. p. 1324–1332. NIPS'10 (2010)
6. Sabathiel, S., McClelland, J.L., Solstad, T.: Emerging representations for counting in a neural network agent interacting with a multimodal environment. vol. ALIFE 2020: The 2020 Conference on Artificial Life, pp. 736–743 (07 2020). https://doi.org/10.1162/isal_a_00333
7. Walach, E., Wolf, L.: Learning to count with cnn boosting. In: Proceedings of the 14th European Conference on Computer Vision, part II. vol. LNCS 9906, pp. 660–676 (10 2016). https://doi.org/10.1007/978-3-319-46475-6_41
8. Wynn, K.: Children's understanding of counting. Cognition **36**(2), 155–193 (1990). https://doi.org/10.1016/0010-0277(90)90003-3

CountingSim: Synthetic Way To Generate a Dataset For The UAV-view Crowd Counting Task

Bartosz Ptak[1][0000−0003−1601−6560] and Dominik Pieczyński[1][0000−0003−0275−5629]

Faculty of Control, Robotics and Electrical, Engineering, Institute of Robotics and Machine Intelligence, Poznań University of Technology, 60-965 Poznań, Poland

Abstract. Recent advances in deep learning-based image processing has enabled significant improvements in multiple computer vision fields, with crowd counting being no exception. Crowd counting is still attracting research interest due to its potential usefulness for traffic and pedestrian stream monitoring and analysis. This paper considers a specific case of crowd counting, namely counting based on low-altitude aerial images collected by an unmanned aerial vehicle. In this field, the data scarcity quickly becomes an issue, making training of complex models infeasible. We show that creating synthetic dataset with developed simulator and using it for pretraining results in better performance when benchmarked with DroneCrowd dataset.

Keywords: Deep learning · Crowd counting · Synthetic data generation

1 Introduction

The progress of deep machine learning methods enables the development of more complex algorithms and allows exploiting them for Unmanned Aerial Vehicles (UAVs) tasks. Several solutions are being created to improve human work, e.g. infrastructure monitoring [6, 3] and plant crops [2, 5] analysis. UAVs are also adopted for usage in a crowd counting task. In contrast to CCTV cameras, the footage obtained from the drone point of view is generally more challenging to analyze and demands more complex models. On the other hand, it allows to observe a wider area, even in a difficult and dynamic environment. Unfortunately, the availability of public datasets for the task is not extensive. While collecting drone imagery poses no significant difficulties, the labelling process is a time-consuming task. The largest dataset – DroneCrowd [9] contains 112 video clips (33,600 frames). This number may prove to be too small to train sophisticated neural network models, but labelling it required placing over 4.8 million hand-written annotations. In contrast, Kinetics 700-2020 dataset [7], which is widely used as a benchmark for action recognition algorithms, contains at least 700 video clips for each of the 700 action classes (490,000 clips at minimum).

In this paper, we propose bridging this gap with a simulator generating synthetic UAV data. Our environment consisting of urban and green areas allows for the massive generation of moving pedestrians footage.

In the next sections, we describe the main contributions of this work, including a description of the simulator and the statistics of the generated data. Then, we present obtained results and discuss to whom the dataset is targeted.

2 Simulation Environment and Data Acquisition

In the research, we use the Unity [1] game engine to develop a simulation to generate a synthetic UAV-view crowd counting dataset. We choose this specific platform due to its better compatibility with Linux operating system when compared with Unreal Engine. Another important point is its ability to interface with Robot Operating System. It enables, in particular, the streaming of images from the simulator and active control of the UAV. For the task of capturing visual images (Red-Green-Blue), we utilise the Perception [4] add-on that accelerates the process and offers an extensible toolset for annotations.

The simulation consists of the city map that represents mostly the urban environment: the downtown, squares and parks. We place the navigation mesh in the appropriate places of the city and spawn a certain number of people who automatically move along dynamically generated paths between waypoints randomly selected from a predefined list. It allows to generate city-like crowd traffic and defines some unusual cases, i.e. protests or events. The data acquisition is performed by eight cameras behaving as they would if attached underneath an aerial vehicle. Simultaneously, the script registers the people's ground-truth label as the two-dimensional array.

Fig. 1. The left image shows sample from the DroneCrowd dataset with human heads annotated with red dots. The right image demonstrates samples from our synthetic dataset.

2.1 Dataset Specifications

We perform multiple runs of the simulator and generate 65 sequences in total. They are characterised by varying number of people, altitudes and illumination. We split the dataset into train, validation and test sub-collections, consisting of separate sequences. The resulting sizes are 155,203, 16,648 and 16,018 images respectively. Figure 1 shows the comparison of the sample from the DroneCrowd dataset and images generated by the CountingSim simulator.

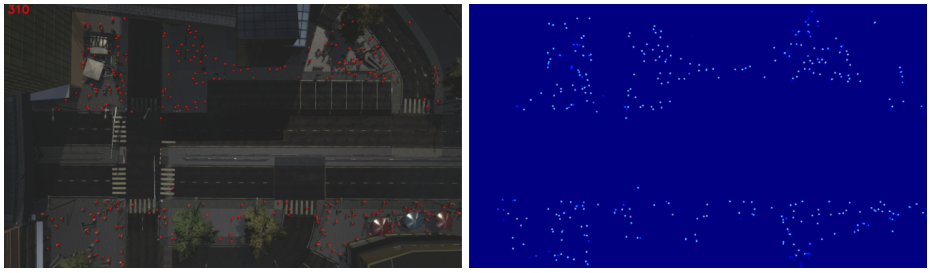


Fig. 2. The sample input image with predicted people poses and summarized number of a crowd (left). Estimated output density mask by a deep neural network (right).

3 Results

The crowd counting task is commonly regarded as a density estimation problem where points corresponding to specific persons are painted on the mask, which is later smoothed using Gaussian filtering. The sum of the image values is equal to the number of people in the crowd. In the paper, we utilise the same approach and develop Unet-like algorithms for the task. Figure 2 illustrates both the post-processing input picture and output density mask.

The metric used in the task is CountingMAE which represents the absolute difference between ground truth and the estimated number of people in the crowd counting task. This metric is not directly relevant to the density estimation task, and it translates into the estimated number of people instead. Table 1 shows improvement of metrics for the DroneCrowd dataset. The training process that includes freezing pre-trained encoder usually results in superior performance when compared with unfrozen encoder. This can be attributed to the overfitting of the models to the training dataset.

4 Discussion

In this paper, a synthetic way to generate a dataset for the crowd counting task is provided. Our experiments show that utilizing synthetic data in a pretraining scenario can boost the final metrics of models and improve the training process

Table 1. The comparison of various models with different weights initialisations and freezing methodologies.

Encoder	Initial weights and training mode [CountingMAE]			
	ImageNet (full training)	CountingSim (full training)	CountingSim (freeze encoder)	CountingSim (freeze encoder and decoder)
resnet18	25.049	23.504	**23.048**	28.295
resnet34	22.053	22.037	**20.214**	56.272
semnasnet_075	20.385	22.026	**17.753**	63.828
semnasnet_100	21.561	**21.307**	22.639	149.15
efficientnet-b0	24.684	28.796	**22.984**	74.165

stability. Similar effects are reported in [8], where the usage of synthetic data has improved the metrics in the CCTV crowd counting task. One of the notable differences between this work and related synthetic datasets in the crowd counting task is the UAV-view characterisation of captured images. However, the used game engine provides tools for light and weather conditions manipulation, which enables generating more near-realistic data.

In the future, we plan to generate more synthetic data in various environments and share the CouningSim dataset publically available. Moreover, we want to measure the transfer learning impact for state-of-the-art models and attempt to apply style-transfer to make the overall look of the generated images more realistic.

References

1. Unity - game engine. https://unity.com/, accessed: 2022-02-20
2. Banerjee, B.P., Sharma, V., Spangenberg, G., Kant, S.: Remote Sensing **13**(15) (2021). https://doi.org/10.3390/rs13152918
3. Banic, M., Miltenovic, A., Pavlović, M., Ćirić, I.: Intelligent machine vision based railway infrastructure inspection and monitoring using uav. Facta Universitatis, Series: Mechanical Engineering **17**, 357 (11 2019). https://doi.org/10.22190/FUME190507041B
4. Borkman, S., Crespi, A., Dhakad, S., Ganguly, S., Hogins, J., Jhang, Y., Kamalzadeh, M., Li, B., Leal, S., Parisi, P., Romero, C., Smith, W., Thaman, A., Warren, S., Yadav, N.: Unity perception: Generate synthetic data for computer vision. CoRR **abs/2107.04259** (2021)
5. Donmez, C., Villi, O., Berberoglu, S., Cilek, A.: Computer vision-based citrus tree detection in a cultivated environment using uav imagery. Computers and Electronics in Agriculture **187**, 106273 (2021). https://doi.org/https://doi.org/10.1016/j.compag.2021.106273
6. Peng, X., Zhong, X., Zhao, C., Chen, A., Zhang, T.: A uav-based machine vision method for bridge crack recognition and width quantification through hybrid feature learning. Construction and Building Materials **299**, 123896 (2021). https://doi.org/https://doi.org/10.1016/j.conbuildmat.2021.123896
7. Smaira, L., Carreira, J., Noland, E., Clancy, E., Wu, A., Zisserman, A.: A short note on the kinetics-700-2020 human action dataset. arXiv preprint arXiv:2010.10864 (2020)

8. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8198–8207 (2019)
9. Wen, L., Du, D., Zhu, P., Hu, Q., Wang, Q., Bo, L., Lyu, S.: Detection, tracking, and counting meets drones in crowds: A benchmark. In: CVPR (2021)

Deep Neural Network Interpretability Methods for Supervised and Unsupervised Problems

Andrzej Brodzicki[1][000000017713526X], Dariusz Kucharski[1][0000000201072407], Michał Piekarski[1,2][0000000193914263], Aleksander Kostuch[1][0000−0003−1242−9851], and Joanna Jaworek-Korjakowska[1][0000000301468652]

[1] Department of Automatic Control and Robotics, AGH UST, Krakow, Poland
{brodzicki,kucharski,piekarski,kostuch,jaworek}@agh.edu.pl
https://home.agh.edu.pl/ mdig/dokuwiki/doku.php
[2] SOLARIS National Synchrotron Radiation Centre, UJ, Krakow, Poland

Abstract. In recent years, deep neural networks (DNNs) have experienced a dynamic rise in applicability in many fields, from industry, through social media to healthcare. In this paper we focus on model interpretability for image analysis as it is a crucial point while deploying the methods in real life. We compare three visualisation algorithms including GradCAM, LIME and Occlusion that increase the model interpretability and check if the assessment is based on correct parts of the image or surrounding. We have compared the effectiveness of these methods in four different image processing research areas including 1) dermoscopic image classification, 2) lung nodule segmentation on CT scans, 3) classification of beam images for anomaly detection in synchrotron, 4) classification of seat occupancy. We briefly describe the model interpretability methods, compare achieved results and draw conclusions.

Keywords: Deep neural networks · Interpretability · Explainability · Supervised · Unsupervised

1 Introduction

One of the main advantage of deep learning is the ability to extract important features from raw data, with almost no preprocessing or expert knowledge. Such an approach is able to solve many computer vision tasks which couldn't be achieved with regular machine learning algorithms, however,this also leads to a low level o trustworthiness, as the user often does not understand how the network works and what criteria have been taken into account to draw the final decision. It is especially crucial in high-risk areas such as healthcare or autonomous driving. An answer to this is the idea of explainable and interpretable AI. The first term focuses on making users understand how it works. The second tries to trace the exact path of decision making inside the algorithm itself. Both terms are closely tied, as tools used for interpretability are often used for explanation too. This paper has been organized as follows: in Section 2 we provide a brief description of three state-of-the-art interpretability techniques from the

field of computer vision, in Section 3 we present the achieved outcomes in different computer vision problems including synchrotron beam stability assessment or seat occupancy classification in vehicle interior.

2 Deep learning model interpretability methods

2.1 Related works

Q. Xhang et al. described four groups for interpretability of convolutional neural networks (CNNs): filters visualisation as a most direct way to explore patterns hidden inside a neural unit, pattern retrieval based on extraction of mid-level features (conv-layers), model diagnosis by checking image regions accountable for the prediction (i.e. gradient-based methods, LIME) and finally, learning a more meaningful representation right from its design [8]. Hedstrm et al. introduced a versatile tool for quantification of a wide range of evaluation metrics [1].

2.2 Interpretability algorithms description

We have chosen three popular model interpretability algorithms which are mostly based on visualization of CNNs activities: GradCAM, LIME and Occlusion. Gradient-weighted Class Activation Mapping (GradCAM) utilizes the gradients of a target (i.e image of a certain class in a classification network) fed into the last convolutional layer to produce a map highlighting the important regions in the image for predicting this particular class [6]. Local Interpretable Model-agnostic Explanations (LIME) explains the predictions of a classifier by learning an interpretable model locally around the prediction [5]. Finally, Occlusion map is performed by analysing the classifier output by occluding portions of the input image, showing which parts of the scene are important for classification [7].

3 Experimental comparison of interpretability methods

3.1 Melanoma malignancy classification in different anatomic sites

Diagnosing a melanoma (a deadly skin cancer) is a challenging task even for expert dermatologists. There is a need for decision-making systems to assess the variety of morphological arrangements in skin moles. In paper [2] we focused on classifying skin lesions originating in different anatomic sites of the body, with a 97% ACC. We also proposed a statistical metric, based on the overlap of Grad-CAM heatmaps and the segmentation ground-truth, to quantify the interpretability (Fig. 1).

3.2 Nodule detection on patches extracted from lungs CT scan

The objective of this research is to localize a nodule change on lungs CT scan images. First step of an algorithm is to extract patches from a single scan and

asses if they contain a nodule change. Fig 1 shows that trained neural network is focusing on finding a nodule change on such patches - GradCAM and Lime show network's attention focused on a nodule itself and in occlusion case one can notice negative predictions if areas around nodule were occluded.

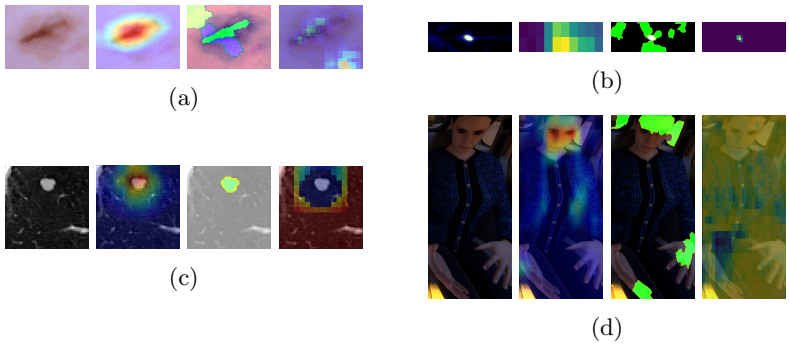


Fig. 1: DNN attention areas obtained from interpretability algorithms - from left to right: input image, GradCAM, LIME and Occlusion results for: a) melanoma classification, b) anomaly detection in synchrotron, c) lung nodule detection, d) seat occupancy classification. The more intense the color, the greater the attention of the neural network.

3.3 Electron beam anomaly detection in Pinhole diagnostic line

Detection of anomalies and instability in diagnostic signals in the SOLARIS synchrotron, with particular emphasis on the images of transverse electron beam profile from the Pinhole diagnostic beamline [4], allows operator to better tune crucial elements of the storage ring (i.e 3rd harmonic cavities) and to observe the state of the entire system. In such complicated and distributed systems, detecting unwanted events and understanding them prevents financial losses, unplanned downtime and damage to the infrastructure. Therefore it is crucial to build interpretable models. Our model results (94% ACC) are shown in Fig. 1.

3.4 Seat occupancy classification in vehicle interior

Classification of seat occupancy in in-vehicle interior is a promising area in new generation cars. In our study [3], we provided an interpretable solution (using ResNet, DenseNet, EfficientNet) that identify object parts without direct supervision. We also proposed two new statistical metrics based on the multivariate Gaussian distribution in order to assess heatmaps without using human-labeled objects. We demonstrated that our interpretability results correlate with the accuracy and can be implemented to work with any resolution for various applications (Fig. 1). Extensive experiments were carried out on 7,500 BMW X5 images from SVIRO database. The model achieved a state-of-the-art result (79.87%

ACC, 95.92% recall, 90.32% spec.) in the domain of seat occupancy classification into seven main categories: empty infant in infant seat, child in child seat, adult, everyday object, empty infant seat and empty child seat.

4 Conclusions

In this work, we have proved importance of the interpretability of the DNN models by testing and comparing different methods in various tasks. During the analysis, most of the results were in line with expectations, but we observed individual cases where the concentration of the network falls on the area that is actually the background for a significant object. With this knowledge, we can consider such a result not entirely reliable, even if it is positive, which allows us to improve the operation of the model by supplementing the data set with similar examples. In summary, the conclusions drawn from the tested solutions allow not only to understand and explain the operation of the models, but also to improve their final reliability. Further research efforts will in the first place be directed to the integration of various methods of interpretability in order to find a method that is even more versatile in terms of the problem being solved.

Acknowledgments

We gratefully acknowledge the funding support of the "Excellence initiative—research university" programme for the AGH UST.

References

1. Hedström, A., et al.: Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. ArXiv (2022)
2. Jaworek-Korjakowska, J., Brodzicki, A., Cassidy, B., Kendrick, C., Yap, M.H.: Interpretability of a deep learning based approach for the classification of skin lesions into main anatomic body sites. Cancers **13** (2021)
3. Jaworek-Korjakowska, J., Kostuch, A., Skruch, P.: Safeso: Interpretable and explainable deep learning approach for seat occupancy classification in vehicle interior. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
4. Kisiel, A., Marendziak, A., Ptaszkiewicz, M., Wawrzyniak, A.: X-ray pinhole camera for emittance measurement in solaris storage ring (05 2019)
5. Ribeiro, M.T., et al.: Why should i trust you?: Explaining the predictions of any classifier. ACM KDD: Knowledge Discovery and Data Mining (2016)
6. Selvaraju, R.R., et al.: Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization (2016)
7. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
8. Zhang, Q., Wu, Y.N., Zhu, S.C.: Interpretable convolutional neural networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)

Detecting Clashes in Boxing

Piotr Stefański[1][0000−0003−1229−327X]

Department of Machine Learning,
University of Economics in Katowice, 1 Maja, 40-287 Katowice, Poland
piotr.stefanski@ue.katowice.pl

Abstract. Sports video analysis is currently an important issue from the point of view of both statistics and live sports coverage. In the context of boxing, computer vision can be used, among other things, to present statistics and to indicate whether there is a contact between the boxers at a particular moment. In boxing, the situation where there is a contact between the boxers does not last long for most of the fight. In this paper, we proposed a solution by which contact between the boxers is detected. This is realised by detecting the boxers and then determining the Euclidean distance between them. The solution was tested on real data, video recorded by us during a boxing tournament. The results obtained show that it is possible to detect contact during boxing and thus skip the rest of the fight. We showed that the contact between boxers occurs in less than a half of the overall time of a fight.

Keywords: clash detection · video processing · combat sport analysis.

1 Introduction

These days computer vision plays a crucial role in the world of sports. Currently dozens of cameras in stadiums are tracking players and balls for game analysis. Computer vision provides knowledge that has a wide range of stakeholders. Beginning with spectators, television presenters, referees and ending with coaches who analyse entire games and the performance of individual players [3, 7, 10, 13].

Competitions in many sports are analysed by advanced and accurate systems every day. Image processing plays important role in these solutions. Computer vision is being used by modern systems to tracking cricket balls and generate a trajectory in 3D based on up to ten high speed cameras with mean error of 2.6 mm [9]. However, current research on combat sports video is very limited, mainly in boxing [5].

By this time scientific community has proposed several approaches to analyse combat sports like boxing. One of them basing on wearable sensors [4, 11, 12] that may be dangerous to players or banned by game regulations, as in NBA games [9]. Tag-based systems also have a problem of short battery life, which is up to 4 hours on some systems [9].

Instead of wearable devices and sensors approach it is possible to measure features of performance using computer vision. The mentioned method is called

as non-invasive approach to analyse pugilists in the ring. It is possible to track boxing gloves using footage from one static frontal view camera [10]. Methods using depth data are also used for tracking boxers and classifying punches; authors [1, 5] chose overhead camera position to reduce problem with occlusions (by another boxer or referee in boxing ring).

The purpose of this paper is to prepare a framework for detect clashes, which is based on frontal view cameras in each corners in a boxing ring. Collecting the relevant footage for this purpose was necessary. Therefore the authors were recording real boxing bouts and prepared the complete training set.

The remainder of the article is organized as follows. Section 1 contains an introduction to current computer vision responsibility in sports and a review of several approaches by other authors. Section 2 contains the proposed approach to detect clashes in boxing. Section 3 contains details of the data collection process and experiments results. The last Section 4 contains summary, conclusion of this paper and further works.

2 Clash Detecting

Boxing competitions characterizes with a lot of breaks between fights and rounds. When watching footage of the entire competition, it appears that for a long time nothing happens in the boxing ring. Clash detecting is the preprocessing step before detecting and classifying blows on the footage. For about 70% of the time (in the footage of the whole competition) boxers are not engaged in close-combats situation according to our observations. Clash detecting is designed to extract interesting parts of footage, where boxers are in close-combats situations. Filtered material is ready for detection and classification of boxer punches.

Boxing is a highly dynamic sport which requires high resolution and high frame rate cameras to capture the details. Using such high quality generates a huge video file which leads to the big computational complexity. An approach to clash detection that reduces the volume of data processed is extremely necessary.

Clash detecting framework based on the authors' approach for boxer detecting (See [8]). Boxer detecting based on person detection [2, 6] with color based approach to filter judge and people outside the boxing ring.

Euclidean measure from equation (1) is used to calculate distance between boxers in the ring. The algorithm detects a clash while distance between the detected boxers is close. A clash is a potential situations where blows could occur between boxers.

$$d\left(A,B\right)=\sqrt{\left(x_B-x_A\right)^2+\left(y_B-y_A\right)^2}, \tag{1}$$

where A and B are points in the coordinate system, x and y stand coordinate for values of the points.

3 Experiments

The necessary footage of the boxing bouts was recorded in Poland at the Silesian league games for juniors, cadets and seniors. For this purpose four GoPro Hero8 cameras with power banks and 128 GB memory cards were used. The cameras were mounted behind each corner on 1.8m high tripods and were recording video in full HD resolution at 50 frames per second. After the competition, which lasted four hours, each memory card was nearly full, totaling just under 500 GB of recorded footage.

A 12 minutes long footage containing 35,000 frames was selected for the experiments. Footage includes 3 rounds of one fight and one round of the next fight. Between rounds 3 and 4 several clashes were detected as shown in the Figure 1. Between fights competitions greet each other, thank each other and receive awards, therefore a few clashes can be detected in that time.

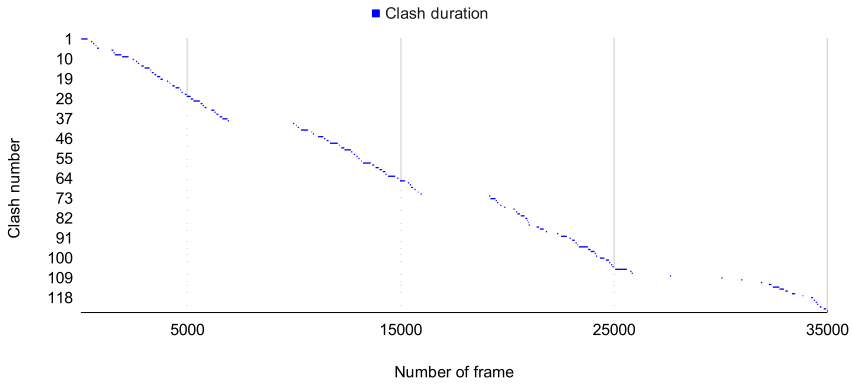


Fig. 1. Diagram of the detected clashes in the footage

The Figure 1 shows detected clashes in the video recording. In the selected footage clashes being were detected in 14,600 frames, which means that for 41% of the recording boxers are in close-combats situation. In this way, the remainder of 59% of the footage can be filtered out. Gaps on the graph represent breaks between rounds and situations where boxers are not close to each other.

4 Conclusion

The purpose of this paper was to propose the approach to detect clashes in boxing bouts footage. Important part of the work was providing a protocol of recording and collecting relevant footage of boxing fights. The authors set up a recording environment and chose special equipment for this purpose.

The experiments confirm that clash detection framework can be used in preprocessing part which allows to reduce large volume of data. In the future authors will synchronize the video from all cameras to reduce problem with occlusions.

References

1. Behendi, S.K., Morgan, S., Fookes, C.B.: Non-invasive performance measurement in combat sports. In: Proceedings of the 10th International Symposium on Computer Science in Sports (ISCSS). pp. 3–10. Springer (2016)
2. Braun, M., Krebs, S., Flohr, F., Gavrila, D.M.: Eurocity persons: A novel benchmark for person detection in traffic scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 1844–1861 (2019). https://doi.org/10.1109/TPAMI.2019.2897684
3. D'Orazio, T., Leo, M.: A review of vision-based systems for soccer video analysis. Pattern recognition **43**(8), 2911–2926 (2010)
4. Hahn, A., Helmer, R., Kelly, T., Partridge, K., Krajewski, A., Blanchonette, I., Barker, J., Bruch, H., Brydon, M., Hooke, N., Andreass, B.: Development of an automated scoring system for amateur boxing. Procedia Engineering **2**(2), 3095–3101 (2010). https://doi.org/https://doi.org/10.1016/j.proeng.2010.04.117, https://www.sciencedirect.com/science/article/pii/S1877705810003711, the Engineering of Sport 8 - Engineering Emotion
5. Kasiri-Bidhendi, S., Fookes, C., Morgan, S., Martin, D.T., Sridharan, S.: Combat sports analytics: Boxing punch classification using overhead depthimagery. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 4545–4549. IEEE (2015)
6. Kundid Vasić, M., Papić, V.: Multimodel deep learning for person detection in aerial images. Electronics **9**(9), 1459 (2020)
7. Li, H., Tang, J., Wu, S., Zhang, Y., Lin, S.: Automatic detection and analysis of player action in moving background sports video sequences. IEEE transactions on circuits and systems for video technology **20**(3), 351–364 (2009)
8. Stefański, P., Kozak, J., Jach, T.: The problem of detecting boxers in the boxing ring. Submitted to the ACIIDS2022 conference (2022)
9. Thomas, G., Gade, R., Moeslund, T.B., Carr, P., Hilton, A.: Computer vision for sports: Current applications and research topics. Computer Vision and Image Understanding **159**, 3–18 (2017)
10. Wattanamongkhol, N., Kumhom, P., Chamnongthai, K.: A method of glove tracking for amateur boxing refereeing. In: IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005. vol. 1, pp. 6–9. IEEE (2005)
11. Worsey, M.T.O., Espinosa, H.G., Shepherd, J.B., Thiel, D.V.: An evaluation of wearable inertial sensor configuration and supervised machine learning models for automatic punch classification in boxing. IoT **1**(2), 360–381 (2020). https://doi.org/10.3390/iot1020021, https://www.mdpi.com/2624-831X/1/2/21
12. Worsey, M.T., Espinosa, H.G., Shepherd, J.B., Thiel, D.V.: Inertial sensors for performance analysis in combat sports: A systematic review. Sports **7**(1) (2019). https://doi.org/10.3390/sports7010028, https://www.mdpi.com/2075-4663/7/1/28
13. Zhu, G., Xu, C., Huang, Q., Rui, Y., Jiang, S., Gao, W., Yao, H.: Event tactic analysis based on broadcast sports video. IEEE Transactions on Multimedia **11**(1), 49–67 (2008)

Evaluation of an Embedded Computer Vision System for Outdoor Media Audience Analysis

Bogusław Cyganek[1,2], Piotr Janyst[1], Mateusz Knapik[2], Łukasz Przebinda[1], Marta Ptaszkiewicz[1], and Tomasz Tajmajer[1]

[1] MyLED, Cracow, Poland {p.janyst, l.przebinda, m.ptaszkiewicz, t.tajmajer}@myled.pl
[2] AGH - University of Science and Technology, Cracow, Poland {cyganek, mknapik}@agh.edu.pl

Abstract. As digital advertisement screens become more popular, there is a need of providing technologies for monitoring the audience of such media. We present the ARA system, which consists of a hardware platform for GPU-accelerated computer vision and a SaaS platform for data analysis. We conducted experiments to evaluate the accuracy of the system in pedestrian counting scenario as compared to a human observer. In this paper we present architecture of this system, as well as the results of its evaluation.

Keywords: Pedestrian monitoring· Machine Learning · Audience analysis, Digital out-of-home media

1 Introduction

Recent growth of digital-out-of-home (DOOH) market result in higher demand for physical audience analysis technologies. The stakeholders are used to the customer data and ease of targeted marketing available in the main digital medium - the world wide web. Similar expectations arose for new kinds of media that are no longer purely digital - such as outdoor screens, interactive displays or augmented reality. Here we present a technology which aims to provide physical audience monitoring - dedicated for use with the DOOH media. The proposed ARA system is a mix of an AI-enabled computer vision hardware with an analytic SaaS platform. The basic functionality of the platform is to provide data about the number of people moving in front of a digital advertisment, and thus enable advertisment providers to estimate their audience volume. Since the system is already at production level, we managed to perform an internal evaluation of its functionality, to prove its effectiveness and create a baseline for further improvements.

In this work we focus only on pedestrians in close range to the advertisement medium - the full audience consists of people in vehicles such as cars and busses, as well as non-pedestrians e.g. people sitting in cafes and restaurants, yet those are out of scope of this evaluation.

1.1 Related Works

The modern ARA system, described in this paper, is based on the ARAHUB platform presented in [7]. The new ARA system was created to improve mostly on the outdoor scenarios. In particular, the imaging sensor was re-designed to provide a wide field of view by using two cameras. ARA sensors were also modified to be standalone devices dedicated for observing a single point of interest.

Similar human audience counting systems were investigated in [1,5,3,2]. Commercial solutions from companies like Quividi or Aquaji exist on the market, however detailed descriptions of their accuracy and evaluation methodology are not publicly available for comparison. A similar embedded system, but for the rescue purposes, was proposed by Gąsienica-Józkowy et al. [4]. Its main component is a weighted YOLOv3 based detector trained with silhouettes of floating persons for rescue reasons. On the other hand, an underwater detection system based on the SSD detector was proposed by Knapik et al. [6].

2 System architecture

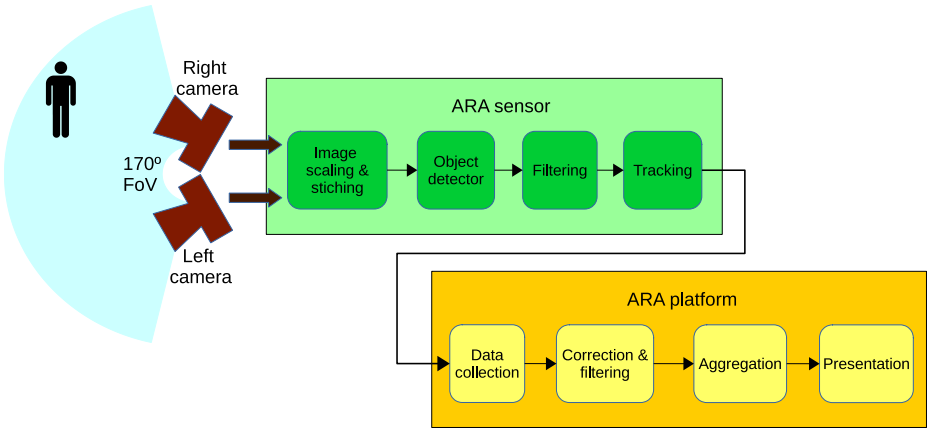


Fig. 1. ARA system architecture: images from two cameras are processed by ARA sensor to detect objects. Movement paths of humans are transmitted to ARA platform for storage, processing and presentation.

The ARA system, presented in Figure 1, consists of two main parts: a) a network of independent embedded devices with GPU acceleration and equipped with a pair of cameras (ARA sensor); b) a central service for data storage, processing and presentation (ARA Platform). The ARA sensor is based on NVidia's jetson nano - an embedded ARM platform equipped with CUDA-capable GPU accelerator, dedicated for use in ML applications. The device includes two HD

cameras, which are set side-by-side in order to observe a FoV with the 170° horizontal and 110° vertical angles, respectively.

Images from these two cameras are stitched together using the cylindrical projection to form a single panoramic image. Next, the image is resized to 384x160 pixels and objects are detected using YOLOv4-CSP [8]. The network is retrained on a custom dataset for improved person detection in both day and night lighting conditions.

Objects that do not meet conditions regarding size and certainty are filtered out. A tracking algorithm based on the Kalman filter is applied to objects detected in subsequent frames to estimate their movement paths. Such paths are then transmitted to the ARA platform.

ARA platform receives and stores movement paths from a large number of devices mounted in different localizations. The data is conditioned and filtered to remove paths that are too short, anomalous or fall out of manually defined areas of interest. Objects are then counted and aggregated statistics for various time periods are calculated. Finally, the platform provides web interface for presenting the data in an informative visual form.

3 Evaluation

3.1 Methodology

In order to evaluate the accuracy of the ARA system for pedestrian count we have prepared a custom dataset that consists of fragments of video footage recorded by the ARA sensor. To obtain ground truth values - the real number of pedestrians - the dataset was manually annotated by a group of researchers. For each footage an area of interest was defined - both human annotators and the ML model counted persons passing through that area, neglecting other visible objects. The dataset was split randomly and uniformly between 8 annotators, each annotated both day and night recordings. Next, the same dataset was processed by the model used in the ARA-sensors. Then, detected movement paths were parsed by the ARA-platform and the resulting pedestrian counts were stored.

3.2 Results

The results are presented in Table 1. Recorded samples were taken from 18 locations during the day and night in different weather conditions and different distances from the camera. The density of pedestrians vary in the selected locations. The dataset consisted of 207 minutes of real-time recordings. In total, almost 1000 pedestrians were visible. The total number of pedestrians detected by the system is close to the manually counted value, however the system has noticeable variance. It is worth noticing that we do not know the manual counting error. In locations with perfect observation conditions, the error can be as low as 5% during the day and 7% at night. In a few locations the system was working in almost complete darkness, yet it was capable of counting at least some pedestrians.

Table 1. Results for pedestrian count for 18 locations in both day and night conditions. The total number of pedestrians are presented for manual and automated count

	Locations number	Manual count	Estimated count	Estimated count MAPE		
				Mean	9-th decile	1-st decile
Day conditions	18	534	559	13.8%	5.3%	30.5%
Night conditions	18	407	424	29.8%	7.3%	62.8%

4 Conclusions

In this paper we have shown the evaluation results of the ARA system - in particular its capability to count human audience in the outdoor digital advertisement conditions. The accuracy of the AI system based on DNN, running on an inexpensive embedded system, is comparable to human results in the daylight scenarios and slightly worse in the nighttime conditions. As the system is not universally applicable yet, it may provide useful data when used in the right environment.

Acknowledgments

This work was supported by the National Centre for Research and Development, Poland, under the grant no. POIR.01.01.01-00-1116/20.

References

1. Brunetti, A., Buongiorno, D., Trotta, G.F., Bevilacqua, V.: Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. Neurocomputing **300**, 17–33 (2018)
2. Cyganek, B., Woźniak, M.: Tensor-based shot boundary detection in video streams. New Generation Computing **35**(4), 311–340 (Oct 2017)
3. F. Fleuret, J. Berclaz, R.L., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 267–282 (2008)
4. Gąsienica-Józkowy, J., Knapik, M., Cyganek, B.: An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance. Integrated Computer-Aided Engineering **28**, 221–235 (2021), 3
5. Iguernaissi, R., Merad, D., Aziz, K., Drap, P.: People tracking in multi-camera systems: a review. Multimedia Tools and Applications **78** (09 2018)
6. Knapik, M., Cyganek, B.: Evaluation of deep learning strategies for underwater object search. In: First Int. Conf. on Societal Automation (SA). pp. 1–6 (2019)
7. Ludziejewski, J., Grad, Ł., Przebinda, Ł., Tajmajer, T.: Integrated human tracking based on video and smartphone signal processing within the ARAHUB system. In: 15th Conference on Computer Science and Information Systems (FedCSIS). pp. 105–114 (2020)
8. Wang, C., Bochkovskiy, A., Liao, H.: Scaled-YOLOv4: Scaling cross stage partial network. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13024–13033. IEEE Computer Society (2021)

Photorealistic Rendering in the Process of Training Deep Neural Networks

Maciej Stefańczyk[1][0000−0001−9948−6319] and Daniel Chmielewiec[1]

Warsaw University of Technology
Institute of Control and Computation Engineering
Nowowiejska 15/19, 00-665 Warsaw, Poland
maciej.stefanczyk@pw.edu.pl

Abstract. This article aims to analyze the possibilities of using photorealistic rendering to create a training dataset for deep neural networks (object detection). Synthetic datasets are compared with the traditional one made by means of photos. The results answer the question of whether synthetic data can substitute photos in that case. The evaluation and advantages of such an approach are presented.

Keywords: object detection · synthetic training data · rendering

1 Introduction

In recent years, there has been a huge shift towards using deep learning methods for all computer vision tasks. With the change towards the data-driven approach, where the model is no longer explicitly defined, there is a need for big training datasets so that the automatic methods can finally come up with an accurate solution. As a result, a lot more human work is needed for the tedious task of collecting the training data: taking the pictures, processing them, and creating the labels takes a lot of time. Moreover, even with the big existing datasets, there is always a scenario in which someone needs to recognize novel objects, not available in those.

The observed speed-up in the field of deep learning is mainly due to the progress in the hardware: general-purpose GPU computing allowed for the calculations to be done in a reasonable time. The same technological hardware progress is a blessing for the computer graphics field. Photorealistic scenes can now be rendered in virtually no time, making the computer games look better every year. This fact and the availability of many 3D modeling and rendering tools opened the possibility of creating synthetic datasets that look as good as the actual photos.

To create the photorealistic rendering, we need models of the objects we want to use. The whole idea of using synthetic data is useless if the process of model acquisition/creation takes much more time than creating the regular dataset. The models of the objects should resemble the geometry of the original objects, as well as should be properly textured. The easiest way of acquiring

the models is to use the existing datasets. One of the most popular sources is the BOP challenge [2], where there are 11 datasets available, each containing (amongst others) the subpart with the 3D models and textures. Those datasets are, however, mostly meant to be used as a benchmark (as the name of the challenge suggests), so it is unlikely that the required objects can be found there.

There are three main aspects of the presented topic. First of all, the models of the objects are required, along with good quality textures and environment scenes. After that, the actual render must be done. For this purpose, either the general-purpose physical-based renderers or the game engines capable of real-time rendering are used (those two distinct approaches are tested in this article). Finally, the evaluation should be done on the real-life datasets to check whether the trained networks can generalize the knowledge to the actual scenario.

2 Experimental setup

Objects for the experiments were selected to cover the popular shapes and sizes, with different texture features, like glossiness or transparency. Models of ten objects were prepared in Blender software, with textures gathered either by scanning existing ones or downloading them from the internet. The mesh complexity differs from very simple boxes, through flat bags, to more complex bottles and cans (fig. 1).

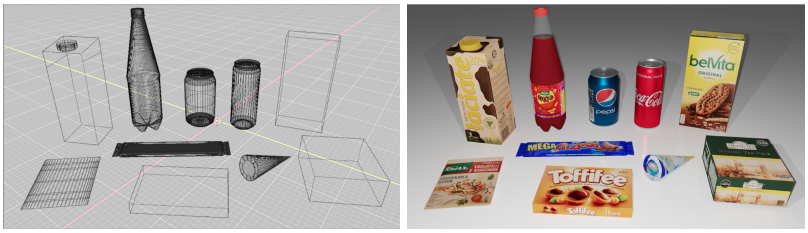


Fig. 1. Preview of the created object models

For the dataset creation process, two popular tools were selected. Both of them outputs the same kind of results, i.e., the RGB image, pixel-perfect segmentation maps, bounding boxes, and others. NVIDIA Deep Learning Dataset Synthesizer (NDDS) is a UE4 plugin from NVIDIA [4]. It takes advantage of the rendering engine in UE4, which is not physically accurate but allows for much faster data generation. The **NDDS** dataset was created using this tool. BlenderProc is an open-source and modular pipeline for rendering photorealistic images of procedurally generated 3D scenes [1]. It uses the Blender as the rendering engine. The pipelines are created using the simple config files, where each step of the process is defined. Images are generated with a physical-based path tracing renderer. It was used to create the **BP** dataset.

For comparison purposes, the third dataset (**photo**) was manually created. For each object, multiple photos were taken (taking into consideration different positions and lighting conditions). From the pictures, objects were cut out and overlaid on different backgrounds in various places (with additional augmentations). Sample pictures from the datasets are presented in fig. 2. Each dataset contained 1000 images per object.

a)  b)  c) 

Fig. 2. Sample images from (a) BP, (b) NDDS, (c) photo dataset

For testing purposes, the additional dataset was created using photos taken in real-life. Three subsets were prepared: easy, with objects clearly visible in good lighting conditions; moderate, with some occlusions, and hard, with objects farther from the camera and worse lighting conditions (fig. 3).

Fig. 3. Sample images from the testing dataset

3 Results and discussion

3.1 Dataset preparation time

The first aspect of the whole process is how long does it take to create every dataset. The preparation of the 3D models and configuring the rendering pipeline took an hour per object on average. Rendering using NDDS achieved around 10 thousand images per hour, BP was ten times slower, averaging 1 thousand images per hour. Because of this high discrepancy in creation time, the smaller BP300 dataset was created, sampling 30% of data from the BP. That made it possible to compare the results based on the similar dataset creation time. In most cases, we don't have hard limits on the dataset creation time, but one can imagine a scenario in which the service robot has to learn a new object, and the time for creating the dataset plays a crucial role here. The photo dataset took 15 hours to take the photos and crop them manually and an additional 1 hour to generate the augmented dataset.

3.2 Detection accuracy

For the main test, the YOLOv3 [3] pretrained on a COCO dataset was selected as a detection network. The training was conducted until convergence on each of the datasets, and the mAP on the testing sets was used as a basis for final comparison. The results are presented in tab. 1. It is important to remember that the training sets were fully synthetic, but the testing set consisted only of natural photos.

Table 1. Detection results (mAP) in different configurations

Testing dataset	NDDS	BP	BP300	Photo
Easy	**0.929**	0.924	0.916	0.926
Moderate	**0.753**	0.724	0.613	0.713
Hard	**0.537**	0.537	0.424	0.483

3.3 Is it worth it?

Synthetic datasets are either on-par or better than manually created ones. Manual datasets, even with a high level of data augmentation, can create only the views of the object that were present on a source photos. Generating data based on 3D models has much more flexibility in both the positioning and lighting conditions. Results achieved using BlenderProc and NDDS are comparable if we use the same number of training pictures. With the current rendering hardware capabilities (like hardware ray-tracing) one can even imagine the *on-the-fly* creation of training data, especially using real-time optimized engines like UE5.

Acknowledgements

M. Stefańczyk is funded by the National Science Centre, Preludium grant no. UMO-2017/25/N/ST6/02358.

References

1. Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H.: Blenderproc. arXiv preprint arXiv:1911.01911 (2019)
2. Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al.: Bop: Benchmark for 6d object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 19–34 (2018)
3. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
4. To, T., Tremblay, J., McKay, D., Yamaguchi, Y., Leung, K., Balanon, A., Cheng, J., Hodge, W., Birchfield, S.: NDDS: NVIDIA deep learning dataset synthesizer (2018), https://github.com/NVIDIA/Dataset_Synthesizer

Playing Cards and Bidding Calls Detection For Automatic Registration of a Duplicate Bridge Game

Piotr Wzorek[1 [0000−0003−3885−600X]] and Tomasz Kryjak[1 [0000−0001−6798−4444]]

Embedded Vision Systems Group, Computer Vision Laboratory,
Department of Automatic Control and Robotics,
AGH University of Science and Technology, Krakow, Poland
{pwzorek,tomasz.kryjak}@agh.edu.pl

Abstract. In this work, the implementation of a playing cards and bidding calls detection system for the automatic registration of a duplicate bridge game is presented. For this purpose, two YOLOv4 deep convolutional neural networks were used. During the training process, which was carried out using an automatically generated dataset, we obtained a detector characterised by more than 99.9% accuracy on the test set. The prepared system has been evaluated on videos collected during bridge competitions and is characterised by high accuracy. The application has been implemented and evaluated on a CPU and an embedded GPU.

Keywords: Deep neural networks · YOLOv4 · Object detection · Duplicate bridge · embedded GPU

1 Introduction

Duplicate bridge is a logic game played with 52 classical cards. It involves 4 players in two pairs and consists of two consecutive phases: auction and play. During the first of these, the players communicate using the so-called bidding calls. In the second phase, the players play one card each in turn (Figure 1).

During the most prestigious competitions, live broadcasts are performed manually by operators. This is a low-complex and repetitive task, so eliminating human involvement can reduce the broadcasting costs and have a positive impact on its accuracy. Therefore, in this paper, we propose an embedded vision system that can be used to automate this process. For this purpose, we used a solution based on the YOLOv4 deep neural network. To be applied in practise, the solution should enable live transmission, that is, process and analyse the video stream in real time, possibly on an energy efficient and compact platform. With regard to that, we have implemented the detector on an embedded GPU (NVIDIA AGX Xavier).

The main contribution of this paper is the proof that the transmission of bridge games can be automated with the use of modern computer vision algorithms. To the best of our knowledge, this is the first scientific publication on this matter.

Fig. 1: Examples of auction (left) and the play phase (right).

The remainder of this paper is organised as follows. In Section 2 the implementation of cards and bid calls detection is described. Optimisation of the detector for an embedded GPU is explained in Section 3. The evaluation of the solution is described in Section 4. The publication ends with a summary and plans for further work on the application.

2 Cards and Bidding Calls Detection

The system should recognise 52 different cards and 38 unique bidding calls. There have been several classical playing card detection methods published so far. The paper [1] from 2010 proposed a system based on a "classical" approach – the Hough transform was used to locate objects. Classification of the proposed areas was carried out using probabilistic methods to compare a set of symbols based on the orientation of their edges or pattern matching and computing correlations.

Later work described solutions based partially or fully on Deep Neural Networks (DNNs). [2] proposes a solution based on the *EfficentDet* network [3]. In the problem of cards detection during poker games considered, the authors' motivation to choose this architecture was the size of the objects relative to the size of the analysed image (0.7% of the dimension of the image).

In this paper, we have decided to use two YOLOv4 networks [4] – one for card detection, the other for bidding calls. After an initial analysis of the problem, we have selected this architecture, as it can be considered as state-of-the-art due to its detection performance (analogous to the possible alternatives), while characterised by a short inference time. The use of the two networks does not have a negative impact on the speed of the application, because at any stage of the game it is not necessary to recognise these elements at the same time.

The database of cards and bidding calls used during the training process was automatically generated and includes cases specific to the game of bridge – we have described the process in a separate publication [5]. We have generated 39300 images of cards and 30000 images of bidding calls using the described method.

We conducted the training process of the YOLOv4 network on 4 *Tesla V100 GPUs* with an input size of $640x640x3$ and a batch size of 64. For the YOLOv4

Table 1: Achieved detection accuracy and throughput for NVIDIA AGX Xavier.

Used model	accuracy [mAP@0.5]	throughput [fps]
YOLOv4 FP32 - cards	99.96%	9
YOLOv4 TensorTR FP16 - cards	99.3%	20
YOLOv4 TensorTR INT8 - cards	97.2%	28
YOLOv4 FP32 - bidding calls	99.98%	9
YOLOv4 TensorTR FP16 - bidding calls	99.6%	20
YOLOv4 TensorTR INT8 - bidding calls	99.3%	28

model, the Mish activation function, the *CIoU* loss function, and regularisation using DropBlok and Class label smoothing mechanisms were selected. Furthermore, MOSAIC augmentation was used for the input images from the training dataset. In this research, we used the Prometheus computing cluster available in the *Academic Computer Centre CYFRONET AGH.*

We applied the transfer learning technique by using network weights adapted to the detection of objects from the *MS COCO* [6] dataset. For both detectors, we obtained high detection performance – 99.96% mAP@0.5 for card recognition and 99.98% mAP@0.5 for bidding calls recognition.

The obtained playing cards detector has an accuracy higher than the alternative proposals in the literature. For example, the solution based on the *EfficientDet-S* network described in [2] has a detection efficiency of 96.9% mAP@0.5 on the test set. However, significant differences between the case considered in the aforementioned publication and the one described should be taken into account – the network is adapted to perform card detection on other datasets.

3 Implementation on Embedded GPU

In order for the detector to be used for real-time game transmission, the trained models were optimised for the embedded GPU using TensorRT tools [7]. For both networks, 16 bit floating-point (FP16) and 8 bit integer (INT8) representations were prepared. Each model was evaluated on a test dataset (Table 1). As the precision of the weight representation in the network model is reduced, a significant decrease in inference time can be observed with only a slight decrease in precision.

4 Towards Automatic Registration of a Duplicate Bridge Game

To evaluate the prepared detector, we recorded a video during a bridge competition. It involved 15 bidding calls (8 of them unique) and 52 cards. We ran three tests. In the first, each frame was subjected to detection using YOLOv4 without optimisation. All bidding calls were correctly identified. However, the

correct estimate of the number of reported passes by the same player proved to be troublesome. This problem was solved by assuming that if a call other than a pass was not recognised in that players' area – that the player passed. The FP16 and INT8 versions also allowed to recognise all bidding calls correctly.

When analysing the play phase, we have obtained the following results:

- FP32 – 46 out of 52 detected correctly (88.5%),
- FP16 – 45 out of 52 detected correctly (86.5%),
- INT8 – 14 out of 52 detected correctly (26.9%).

They indicate that even a very good detector (97-99% accuracy), trained on a large and varied database, does not necessarily perform as reliably in real conditions. This is particularly noticeable with the INT8 version.

5 Summary

The developed playing cards and bidding calls detection system is characterised by high efficiency. The use of an embedded GPU with an optimised FP16 model enables to obtain a real-time solution. This is possible due to the relatively slow pace of the bridge play. Applying some modifications could make it possible to use the system for preparing live broadcasts of bridge competitions.

We plan to continue the development of the system. As part of further work, we intend to thoroughly analyse the detection system, update the training dataset with "problematic" cases and continue the training process, which could significantly improve the quality of detection of elements used during the game. Additionally, supporting the recognition process with game rules analysis should improve the effectiveness of the solution. For example, the obligation to follow suit could be taken into account when recognising cards.

Acknowledgement. The work presented in this paper was supported by the AGH University of Science and Technology project no. 16.16.120.773.

References

1. Pimentel, J. and Bernardino, A.: A Comparison of Methods for Detection and Recognitionof Playing Cards"
2. Chen, Q.: Poker Watcher: Playing Card Detection Based on EfficientDet and SandglassBlock, 2020 11th International Conference on Awareness Science and Technology (iCAST).IEEE. 2020, s. 1–6
3. Tan,Pang R. and Le, Q. V.: EfficientDet: Scalable and Efficient Object Detection, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778-10787, doi: 10.1109/CVPR42600.2020.01079.
4. Bochkovskiy, A., Wang, C. and Liao, H.M.: YOLOv4: Optimal Speed andAccuracy of Object Detection, CoRRabs/2004.10934 (2020). arXiv:2004.1093
5. Wzorek, P. and Kryjak, T.: Training dataset generation for bridge game registration, Zeszyty Studenckiego Towarzystwa Naukowego, ISSN 1732-0925. arXiv:2109.11861
6. Lin, T.: Microsoft COCO: Common Objects in Context, CoRR abs/1405.0312(2014). arXiv:1405.031
7. The TensorRT SDK, https://developer.nvidia.com/tensorrt, Last accessed 7 March 2022

Retrieving Images from Low-bit Representations With Use of Convolutional Neural Networks

Oleksandr Pankiv[0000−0002−5062−5813] and Dariusz Puchala[0000−0001−9070−8042]

Lodz University of Technology, Institute of Information Technology, Lodz, Poland
oleksandr.pankiv@dokt.p.lodz.pl, dariusz.puchala@p.lodz.pl

Abstract. In this paper we present the preliminary results obtained in the task of reversing the operations of image quantization (we take into account optimal Lloyd-Max quantization and uniform quantization with Floyd-Steinberg dithering) and halftoning with use of convolutional neural networks (CNN) based approach and deep learning techniques. In order to verify the effectiveness of the proposed algorithm we made the experimental analysis based on exemplary test images involving popular objective image quality measures in the form of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). Moreover, the resulting images were analyzed by taking into account their subjective visual examination. The results obtained within the experiments are properly commented and the final conclusions are drawn. The experimental analysis allows to state that the proposed CNN based approach allows for quality improvement and as such it can be used in practical applications of image retrieval from their low-bit representations.

Keywords: Image retrieval· dithering· halftoning· CNN

1 Introduction

The following image processing operations: halftoning, dithering and quantization [1], are used in computer systems to reduce the amount of gray shades in images for the tasks of image compression, presentation of images on low-quality displays and image printing. Since such operations highly reduce the amount of information contained in an image, the task of their reveral is in general an ill-posed problem. However, it was shown in [2] on example of halftoning that convolutional neural networks (CNNs) can greatly contribute in this area allowing to obtain better results than classical methods. In this paper we present the preliminary results obtained with the use of CNNs applied to reversing the mentioned operations. The main contributions of our work are: (I) applying CNN with simpler structure than in [2], (II) extending the analysis to quantization and dithering when compared to reverse halftoning addressed in [2].

2 Low-bit image representations

Halftoning - it is the operation of image sampling and its representation in the form of black and white dots of different sizes dependent on the grayscale values present in the specific areas of an image. It is parametrized by the distance between the dots. In our experiments we consider distance values of 4, 6 and 8.

Quantization - in our paper we consider the optimal Lloyd-Max scalar quantization of input image to the given number of 2, 4 or 8 levels of grayscale.
Dithering - is the operation of image quantization with prior addition of pseudorandom noise. The main role of the noise is to reduce the contouring effects. We considered scalar quantization to the following values of 2, 4 and 8 levels.

3 Network architecture

In our experiments we used CNN with the following topology: (I) two-dimensional convolutional layer (Conv2D) with 64 filters of size 7 on 7, (II) Conv2D layer, 32 filters, size 5 on 5 elements, (III) Conv2D layer, 32 filters, 3 on 3 elements, (IV) Conv2D layer, 32 filters, 3 on 3 elements, (V) Conv2D layer, 1 filter, 11 on 11 elements. In layers (I) - (IV) we used ReLU activation function. The last layer used linear activation function. In all layers we used bias. At input and output we expect grayscale images.

4 Experimental results

In our experiments we trained the CNN with 25 face grayscale images of resolution 512 on 512 pixels taken from the dataset [3] by using Adam optimizer and SSIM measure as the loss function. At the input we provided the low-bit representation of an image demanding its full reconstruction to the original grayscale image at output. Another 25 images from the same dataset acted as the training set with additional four standard test images: lake.bmp, lena.bmp, man.bmp and mandrill.bmp. The obtained results for: (I) halftoning with dots distance 4, 6 and 8, (II) Lloyd-Max quantization to 2, 4 and 8 levels, (III) dithering with Floyd-Steinberg method to 2, 4 and 8 levels are presented in Tables 1-3. One should note that in case of face images we show averaged results.

Table 1. Reversing image halftoning operation

Image(s)	PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM
	dist. 4		dist. 6		dist. 8	
faces	31.66	0.730	31.65	0.734	31.47	0.703
lake.bmp	30.40	0.708	30.29	0.698	30.20	0.655
lena.bmp	32.26	0.805	32.34	0.804	31.99	0.771
man.bmp	30.36	0.684	30.34	0.689	30.14	0.625
mandrill.bmp	29.27	0.535	29.26	0.554	29.11	0.432

The results depicted in Fig. 1 enable to observe that the CNN based approach allowed to reverse halftoning operation generating results which can be subjectively stated as more pleasing to the eye of an observer even if they are relatively soft and luck their original crispness and sharpness. The analysis of objective measures (see Table 1) shows results close to 30 dB and higher in case of faces images or Lena image (here even higher than 32 dB). The value of SSIM was higher than 0.800 only in one case of Lena image.

Reversing optimal Lloyd-Max scalar quantization results in images that can be characterized by higher sharpness (see Fig. 2). Although this operation may

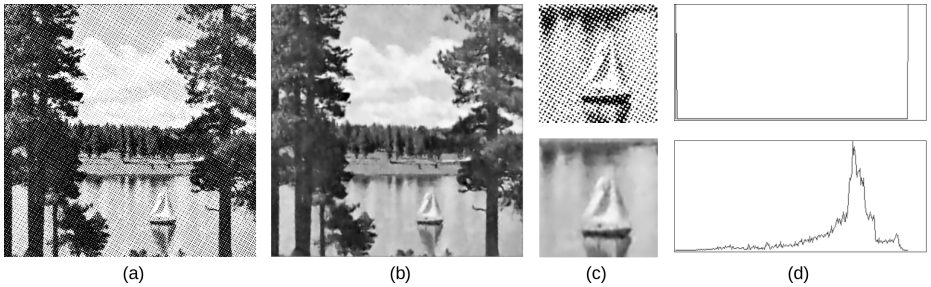


Fig. 1. Results in reverse halftoning for distance between dots equal to 4: (a) halftoned, (b) reconstructed, (c) exemplary fragments of images, (d) histograms of images

make more information to be irrevocably lost (see clouds in Fig. 2). However, the better sharpness is reflected also in the objective measures resulting in higher values of SSIM index while PSNR values do not conform this tendency. It should be noted that PSNR and SSIM measures have different theoretical background.

Table 2. Reversing image Lloyd-Max scalar quantization

Image(s)	PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM
	levels 2		levels 4		levels 8	
faces	28.44	0.605	29.81	0.758	32.42	0.858
lake.bmp	28.33	0.597	29.46	0.763	31.90	0.868
lena.bmp	28.39	0.674	30.37	0.817	33.31	0.889
man.bmp	28.41	0.497	29.38	0.727	31.85	0.874
mandrill.bmp	28.67	0.486	29.51	0.735	31.15	0.889

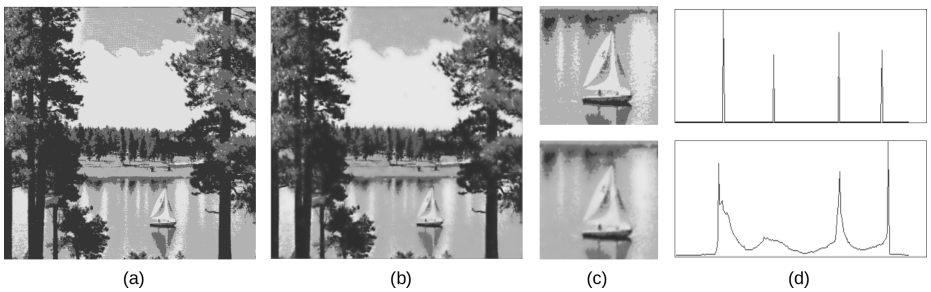


Fig. 2. Results in reversing Lloyd-Max scalar quantization for 4 levels: (a) quantized, (b) reconstructed, (c) exemplary fragments of images, (d) histograms of images

Both subjective and objective results obtained in this part of the experimental analysis prove that reversing the Floyd-Steinberg dithering operation allows to obtain the best results among the considered cases. It can be concluded that dithering operation, though uses the same level of shades as optimal Lloyd-Max scalar quantization, allows to convey much more relevant information contained in an image. Here, the obtained PSNR results are higher than 30 dB for all of the tested images (for levels number higher than 2) and the PSNR values for the Lake image are even higher than 36 dB. Also with SSIM index it was possible to

Table 3. Reversing image dithering operation

Image(s)	PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM
	levels 2		levels 4		levels 8	
faces	28.59	0.690	32.43	0.824	34.08	0.870
lake.bmp	29.48	0.757	34.36	0.891	36.41	0.926
lena.bmp	29.03	0.642	31.48	0.839	33.02	0.901
man.bmp	28.87	0.628	30.32	0.817	30.96	0.877
mandrill.bmp	29.51	0.676	32.87	0.797	34.03	0.840

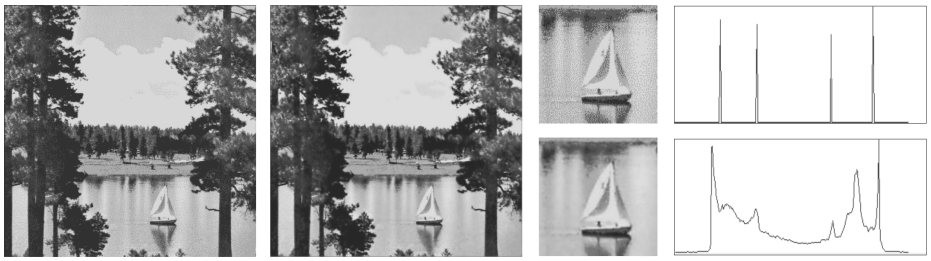


Fig. 3. Results in reversing Floyd-Steinberg dithering for 4 levels: (a) dithered, (b) reconstructed, (c) exemplary fragments of images, (d) histograms of images

obtain higher results even exceeding the level of 0.900 in case of the Lake image. Please note, however, that if one would like to compare the resulting images for the same level of shades of grayscale in all the cases (which is should be 2) then the halftoning operation was the one that keeps the most relevant information related to image details.

5 Conclusions

Based on the conducted experiments it can be concluded that the CNN based approach allows to reverse the operations of image halftoning, scalar quantization and dithering giving the results which can be more adequate for further presentation or archiving of images. This result is solely based on both subjective and objective quality measures. Since the presented results are preliminary we can say that the direction of the future work should be focused on experiments aimed at finding the CNN with the simplest topology understood in the sense of the number of parameters and computational complexity, yet, allowing to obtain the best possible results in image reconstruction.

References

1. Jain A. K., Fundamentals of Digital Image Processing, Prentice Hall, 1989.
2. Hou X., Qiu X., Image Companding and Inverse Halftoning Using Deep Convolutional Neural Networks, arXiv:1707.00116, July, 2017.
3. Karras T., Laine S., Aila T., A style-based generator architecture for generative adversarial networks, CoRR, vol. abs/1812.04948, January 2018.

Thermo Presence: The Low-resolution Thermal Image Dataset and Occupancy Detection Using Edge Devices

Przemysław Aszkowski[1][0000−0003−0388−8546] and Mateusz Piechocki[1][0000−0002−3479−0237]

Faculty of Control, Robotics and Electrical, Engineering, Institute of Robotics and Machine Intelligence, Poznań University of Technology, 60-965 Poznań, Poland

Abstract. Presence monitoring in office buildings is a vivid topic in building management systems. One of the well-established techniques to achieve it is using infrared sensors. In this paper, we present an annotated dataset consisting of low-resolution thermal images from different office rooms, with a changing number of persons in the scene. For each thermal image, a corresponding image from the RGB camera is available for visual inspection. On each thermal image, the centre position of every person is annotated, allowing not only to know the total number of people but also to track their positions. Along with the dataset, an evaluation of U-Net like convolution neural network architecture on low-power edge devices was carried out, with a comparison of their performance and energy consumption. Due to FLASH memory deficiencies on embedded systems, quantization of the models was applied, with an added benefit of shorter interference time. The presented solution allows estimating the presence density map while maintaining low-level power consumption.

Keywords: Thermal Imaging · Deep Learning · Edge AI .

1 Introduction

Office and industrial buildings require a lot of energy to keep comfortable conditions for people, both in terms of lightning and HVAC (heating, ventilation, air conditioning). In majority of buildings there is no presence detection systems or only a binary information, therefore a lot of energy is wasted for conditioning empty rooms. The savings of the energy from using presence sensors are estimated to be around 25% [2]. Identifying people in high-resolution RGB images is a well-established technique, but it raises serious concerns, especially regarding violation of privacy. Thanks to low-resolution and processing on the edge, the presented method allows keeping the privacy of the monitored people, with minimal security risks.

The Thermo Presence dataset presented in this paper consists of annotated low-resolution thermal images, captured from a thermal array sensor (MLX90640) at 0.5 Hz. The dataset can be used to train the models, especially with convolutional neural networks, as applied in [3]. Similar solutions are presented in [1],

where the authors achieved only binary presence information, and [4], but the presented method is both more accurate and resource-efficient. Moreover, both papers did not disclose their datasets.

2 Thermo Presence Dataset Specifications

The dataset consists of thermal images with a resolution of 32x24 pixels, with annotated centres of persons. The data was gathered in four different office rooms, with a changing number of persons from 0 to 5 people on each frame. An exemplary thermal frame presented in 1 shows five people in different positions, with quite significantly varying sizes. All the gathered data were divided into training, validation and test datasets, so that each consists of frames continuous in time, with data split presented in Table 1.

Table 1. The distribution of data in Thermo Presence dataset.

Dataset	**Number of Persons in a Frame**						**Total**
	0	**1**	**2**	**3**	**4**	**5**	
Training	99	105	2984	3217	1953	114	8472
Validation	0	139	631	1691	225	139	2825
Test	162	83	211	341	1235	315	2347

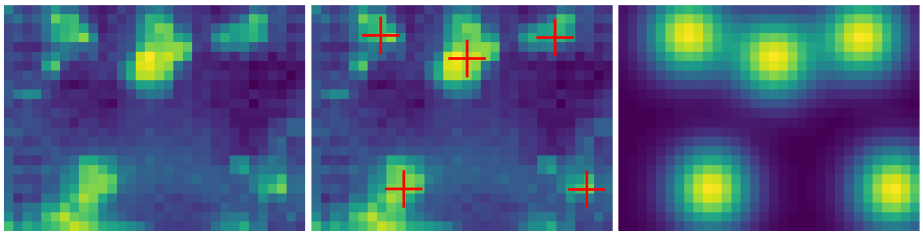


Fig. 1. Thermo Presence dataset: (a) example thermal image from dataset; (b) annotations - person locations; (c) generated density map.

3 Neural Network Architecture

The occupancy counting is a density estimation task and requires an algorithm able to reconstruct the position of persons. Therefore the concept of U-Net architecture [5] was adapted to this task objectives. Although the implemented deep learning model has a simplified, single-channel structure (see Fig. 2), the achieved values of quality measures are very high.

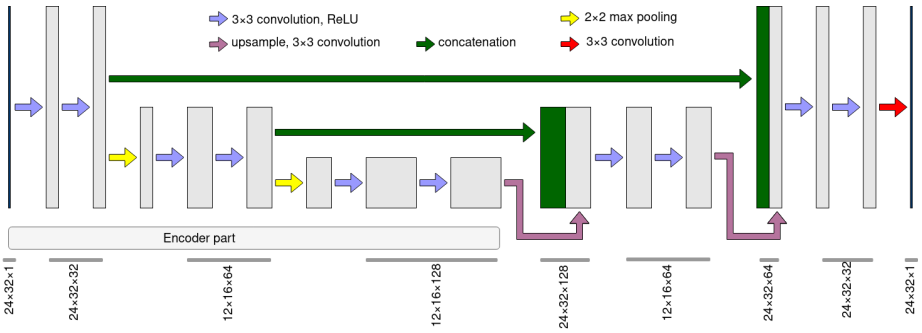


Fig. 2. The proposed U-Net based neural network architecture.

4 Evaluation

The proposed encoder-decoder algorithm was evaluated on Raspberry Pi 4B, a commonly utilized smart home hardware. Moreover, the computing accelerators: Coral USB Accelerator and Intel Neural Compute Stick 2 (NCS2), were also tested. The Coral USB Accelerator is a co-processor with an Edge TPU circuit that natively supports most of the widely used TensorFlow layers. The Intel NCS2 is also a small USB module, designed to accelerate neural networks. Its main part, Intel Movidius Myriad X Vision Processing Unit, is a dedicated hardware neural accelerator.

Table 2. Average inference time and quality measures observed on the tested hardware.

Device	Data Format	Avg. Inference Time [ms]	MAE	MSE	Counting MAE	Counting MSE
Raspberry Pi 4B	FP32	25.2894 ±0.7431	0.1205	0.0493	0.0473	0.0499
Raspberry Pi 4B	FP16	25.3226 ±0.8889	0.1204	0.0493	0.0473	0.0499
Raspberry Pi 4B	INT8	16.4497 ±0.0921	0.4733	0.3459	0.2876	0.3259
Raspberry Pi 4B + Intel Neural Compute Stick 2	FP16	3.0045 ±0.2114	0.1239	0.0507	0.0481	0.0507
Raspberry Pi 4B + Coral USB Accelerator (max)	INT8	0.7107 ±0.0601	0.4729	0.3470	0.2833	0.3208

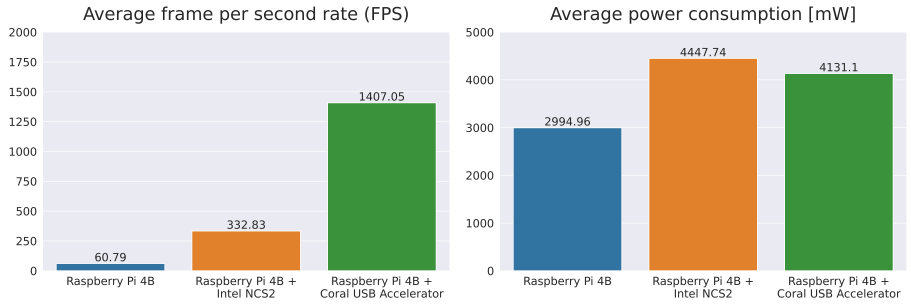


Fig. 3. The algorithm performance measurement: (left) average frame per second rate; (right) average power consumption during inference process.

The hardware evaluation includes inference time measurement and standard regression metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE) calculation. Furthermore, in the benchmark process the Counting MAE and Counting MSE metrics were introduced. This measures compares the output people count in frame instead of its elements one by one. The detailed results are presented in Table 2. In IoT applications, power consumption and efficiency are as crucial as overall algorithm performance. Therefore, the current and voltage were measured on the tested hardware. Figure 3 depicts benchmark results.

5 Discussion

The paper presents a novel dataset for presence detection and occupancy counting approaches. It consists of 13 644 thermal images collected in the office space with added people locations. Moreover, the tiny encoder-decoder neural network was implemented and benchmarked to demonstrate dataset's feasibility. The performance tests have shown that the algorithm based on low-resolution input is able to provide accurate density estimation of people's locations. Furthermore, the method can be used on low-cost, resource-constrained edge devices. Although the Raspberry Pi is able to infer the model, the computing accelerators achieve better performance with only a slight increase in power consumption. The obtained measures indicate that in applications requiring low latency the best results are achieved with the Coral USB Accelerator. On the other hand, if the metrics are as crucial as inference time, the Intel Neural Compute Stick 2 should be considered. The performed tests confirm the usefulness of the dataset and edge AI hardware accelerators in IoT solutions. The Thermo Presence dataset is available in the project's repository at `https://github.com/PUTvision/thermo-presence`.

References

1. Abedi, M., Jazizadeh, F.: Deep-learning for occupancy detection using doppler radar and infrared thermal array sensors. In: Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC). pp. 1098–1105 (2019). https://doi.org/10.22260/ISARC2019/0146
2. Beltran, E., Cerpa: Thermosense: Occupancy thermal based sensing for hvac control. In: In Proceedings of the 5th ACM Workshop on Embedded Systems for Energy-Efficient Buildings. pp. 1–8 (2013). https://doi.org/10.1145/2528282.2528301
3. Kraft, M., Aszkowski, P., Pieczyński, D., Fularz, M.: Low-cost thermal camera-based counting occupancy meter facilitating energy saving in smart buildings. Energies **14**(15) (2021). https://doi.org/10.3390/en14154542
4. Metwaly, A., Queralta, J.P., Sarker, V.K., Gia, T.N., Nasir, O., Westerlund, T.: Edge computing with embedded ai: Thermal image analysis for occupancy estimation in intelligent buildings. In: Proceedings of the INTelligent Embedded Systems Architectures and Applications Workshop 2019. pp. 1–6 (2019). https://doi.org/10.1145/3372394.3372397
5. 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). https://doi.org/10.1007/978-3-319-24574-4$_2$8

Truly Random Color Visual Cryptography without Surplus Color Spikes

Leszek J. Chmielewski[1], Mariusz Nieniewski[2], and Arkadiusz Orłowski[1]

[1] Warsaw University of Life Sciences – SGGW
Institute of Information Technology, Warsaw, Poland
{leszek_chmielewski,arkadiusz_orlowski}@sggw.edu.pl
[2] University of Lodz
Faculty of Mathematics and Informatics, Łódź, Poland
mariusz.nieniewski@wmii.uni.lodz.pl

Abstract. Color visual cryptography with truly random shares is attractive because transferring pure noise conceals the very fact that actually the information is transferred. The previously proposed method exhibits two types of errors, which lead to representing color with loss of brightness and to the presence of spikes with excessive brightness. The method without this second type of error is proposed. The randomness of shares as well as the potential for parallel processing is maintained.

Keywords: Visual cryptography · Color · No spikes · Random shares · True randomness.

1 Introduction

Since the pioneer work of Naor and Shamir [6] new publications in the domain of visual cryptography emerge continually (with [5] as a good example). The subject of randomness in the cryptographic methods is crucial; however, it seems that it is usually studied regarding the random nature of only the coding process (see e.g. [4]). The randomness of the *shares* has also been considered to some extent, for example in [8], where the autocorrelation was used as the measure of randomness. However, the randomness of the substrate in which the coding takes place treated as the means of masking the very process of information transfer has gained little or no interest in the literature so far. Using truly random shares makes it necessary to sacrifice the accuracy of the transferred message to a limited extent, but in the case of image information this is admissible thanks to the capability of our visual system to properly understand the slightly corrupt information. Therefore, in [7] we have proposed the coding of black-and-white images with truly random shares, with their randomness finally confirmed in [3] with the set of NIST randomness tests [1]. Further, this proposition was extended to color images in a series of publications, and finally the actual randomness of the shares for color cryptography was confirmed in [2] with the same tests.

The algorithm of color visual cryptography presented in [2] generates two types of errors in the decoded image. In this paper we propose an algorithm in which only one of these errors can appear.

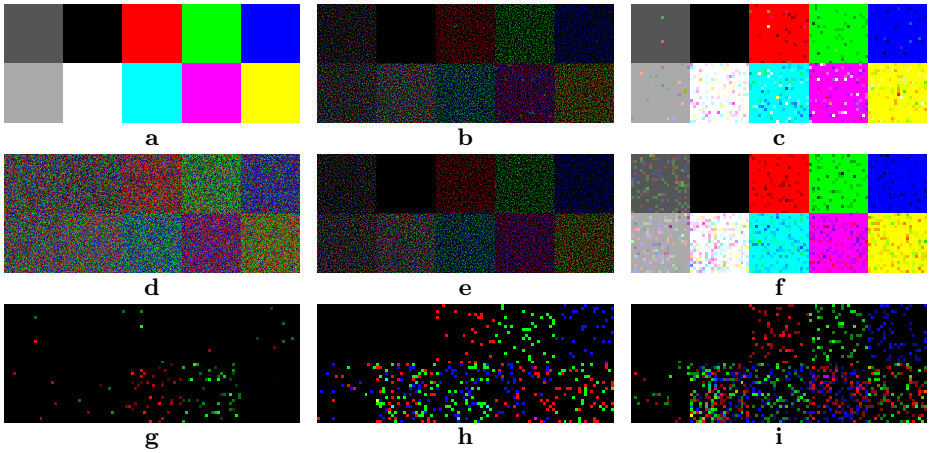


Fig. 1. Decoding, restoration and errors for a test image. (**a**) Secret, 100×40 [3]. (**b, e**) Image **a** decoded 600×240, and (**c, f**) restored 100×40, (**b, c**) with the *classic* method [2], and (**e, f**) with the *no hiding error* method. (**g, h**) Decoding errors 100×40 from *classic* method: (**g**) *hiding failure* error, (**h**) *missing color* error; (**i**) only *missing color* error from *no hiding error* method. Brightest spots in **g**, **h**, **i**: 3 errors per pixel. (**d**) Share from the *switching shares* method, strong information leak visible.

2 The Method

In this short paper we shall only outline the algorithms, and the details can be found in [2,3]. Assume a color R, G, B pixel in the secret image is represented by a 6×6 segment in a share, consisting of randomly displaced R, G, B and black (K) pixels, with equal probabilities. There are two shares. Overlaying one on the other reveals color pixels if colors in a given position are the same (*uncovered*), or black ones, if colors are different (*covered*, ideally narrow-band color filters are assumed). Obtaining a fully bright color (for example, all 36 pixels red and uncovered) in a segment is statistically improbable, so having for example six pixels uncovered in each non-black color is considered as white (brightness loss is unavoidable). Let us call these pixels *open*; numbers of open pixels is established by dithering the secret in a palette of $\{0, \ldots, 6\}$ pixels in each color per segment.

Classic method [2] Initially the shares are equal, so all the pixels are *uncovered*. We attempt to cover all the pixels except the *open* ones by swapping the pixels, by pairs, in share 2 within a segment. If, at random, there are less R pixels than the required number of *open* R pixels in a segment, then the *missing color* error occurs. If some non-*open* pixels cannot be *covered* due to lack of pixels in a covering color in share 2, then the *hiding failure* error occurs.

This is illustrated in Fig. 1b (and c showing the restored image [2]), where for example in a G field some dark pixels appear – *missing color* error, and in a R field some G pixels emerge giving yellow – *hiding failure* error.

No hiding error method (strictly, *no hiding failure error* method) It is proposed to start from shares with all pixels covered: for each pixel in share 1, in share 2

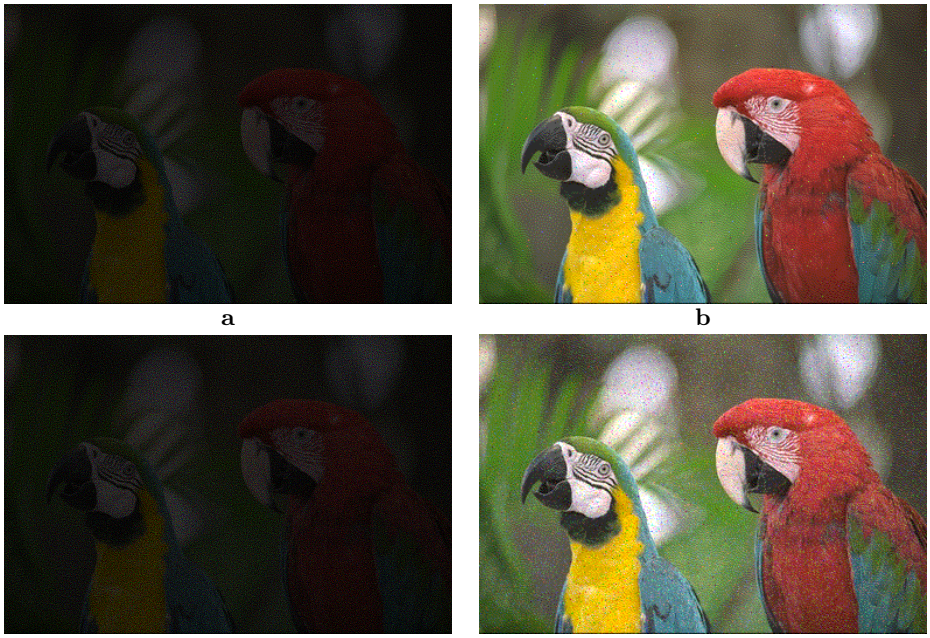


Fig. 2. Results for a natural image `parrots`, 384×256. (**a, c**) Decoded, 2304×1536; (**b, d**) restored, 384×256. (**a, b**) *classic* method [2]; (**c, d**) *no hiding error* method.

a pixel with any color different from that of share 1 is set at random. We attempt to *uncover* the *open* pixels by swapping the pixels, by pairs, in share 2 within a segment, while avoiding *uncovering* any not-*open* pixel. The *hiding failure* error cannot appear, but the *missing color* errors can be statistically more frequent, as there should be enough pixels in a color not in one, but in two different shares. This is illustrated in Fig. 1e, f, where some pixels are too dark – *missing color* error. Unbalanced missing color errors can give rise to changes of hue.

The densities of errors per pixel are larger in general in the *no hiding error* method than in the *classic* method. The total density of two types of errors together from the *classic* method versus the density for one type of error from the *no hiding error* method are: 0.141 vs. 0.191 for the `test100` image, and 0.031 vs. 0.039 for the `parrots` image (Fig. 2). These densities clearly depend on the image brightness, see Fig. 1g-i.

Both above methods need the search for pixels to be swapped which now is performed at random, to keep unchanged the originally random properties of the shares. Speeding up this search is possible, but was not attempted. The processing of each segment is independent and can be performed in parallel.

Switching shares method A method was tried in which the pixel for the share 2 was chosen from share 2 of the *classic* method if the pixel in a segment should be *open*, and from the share 2 of the *no hiding error* method otherwise. The idea would rely on selecting pixels from two random distributions, which should give another random distribution. The catch is that in this case the selection is not

random, but it depends on the secret. The share 2 generated in this way, shown in Fig. 1d, reveals a strong leak of the secret. Therefore, this method, although extremely quick, is totally unsuccessful and will be postponed.

In an example of a natural image (Fig. 2), in spite of that the density of errors is smaller with the *no hiding error* method, the image quality seems to be better with the *classic* method. The *hiding failure* errors which produce surplus color spikes are not harmful in images a, b, while the density of *missing color* errors in the brighter and more colorful regions of images c, d is conspicuous.

The results of randomness tests [1] performed for 100 realizations of coding the images with the new method are similar to those from the *classic* method presented in [2], so in this short paper we shall omit showing the large graphs.

3 Conclusion

A new variant of the color visual cryptography method has been proposed. Only the *missing color* errors appears in it, as compared to the previously used method in which also the *hiding failure* errors are present. The secret images decoded in the proposed method have no surplus bright color spikes, although the color can be more uneven in bright regions. Both the new and the previous method have the same virtue of true randomness of the shares. More research is necessary to speed up the processing, besides that each algorithm can be parallelized.

References

1. Bassham, L.E., Rukhin, A.L. et al.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. (2010), NIST Rep. 800-22 Rev 1a. `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762`
2. Chmielewski, L.J., Nieniewski, M., Orłowski, A.: Can color visual cryptography be truly random? In: Proc. Int. Conf. CORES. LNNS vol. 255, pp. 72–86 (2021). https://doi.org/10.1007/978-3-030-81523-3_7
3. Chmielewski, L.J., Nieniewski, M., Orłowski, A.: Testing the randomness of shares in color visual cryptography. Pattern Analysis & Applications **24**(4), 1475–1487 (2021). https://doi.org/10.1007/s10044-021-00999-5
4. De Bonis, A., De Santis, A.: Randomness in secret sharing and visual cryptography schemes. Theor. Comput. Sci. **314**(3), 351–374 (2004). https://doi.org/10.1016/j.tcs.2003.12.018
5. Liu, Z., Zhu, G. et al.: Contrast-enhanced color visual cryptography for (k, n) threshold schemes. ACM Trans. Multimedia Comput. Commun. Appl. (2022). https://doi.org/10.1145/3508394, accepted Feb 18, 2022
6. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT'94. LNCS vol. 950, pp. 1–12 (1995). https://doi.org/10.1007/BFb0053419
7. Orłowski, A., Chmielewski, L.J.: Generalized visual cryptography scheme with completely random shares. In: Proc. 2nd Int. Conf. Applications of Intelligent Systems APPIS 2019. pp. 33:1–33:6. ACM (2019). https://doi.org/10.1145/3309772.3309805
8. Ulutas, M., Yazici, R. et al.: (2,2)-secret sharing scheme with improved share randomness. In: Proc. 23rd Int. Symp. Computer and Information Sciences ISCIS 2008. pp. 1–5. IEEE (2008). https://doi.org/10.1109/ISCIS.2008.4717857

An Ensemble of Attentive Recurrent Networks with Randomized Dilations for Forecasting

Paweł Pełka[1][0000−0002−2609−811X], Grzegorz Dudek[1][0000−0002−2285−0327], and Slawek Smyl[2][0000−0003−2548−6695]

[1] Electrical Engineering Faculty, Czestochowa University of Technology, Poland
{pawel.pelka,grzegorz.dudek}@pcz.pl
[2] slawek.smyl@gmail.com

Abstract. In this work, we propose an ensemble of recurrent neural networks (RNNs) equipped with attention and dilation mechanisms for forecasting time series (TS) with multiple seasonality. Recurrent cells implement a dilation to better capture complex long-term and seasonal dependencies and an attention for dynamic weighting of input vector components. We stack multiple dilated recurrent layers with hierarchical dilations in order to learn temporal dependencies of different scales at different layers. The novelty of this approach is that additional diversity among learners is generated by randomly delayed connections. The model produces both point forecasts and predictive intervals (PIs).

Keywords: Multiple seasonality · Recurrent NNs · TS forecasting.

1 Introduction

TS forecasting is a challenging problem when a TS expresses multiple seasonality, nonlinear trend and varying variance. NNs can flexibly model complex nonlinear relationships in TS and reflect process variability in uncertain dynamic environments [1]. RNNs, which were designed for sequential data, are extremely useful for TS forecasting [2]. They are able to capture both short- and long-term dynamics thanks to their internal memory and gating mechanism.

In this work, we explore a new RNN equipped with dilation and attention mechanisms, which was proposed in [4]. It was developed for TS with multiple seasonality. To improve the accuracy and generalization property of the model, we propose ensembling with additional diversity among learners generated by randomly delayed connections.

2 Forecasting Model

Forecasting problem. We formulate the forecasting problem given a length n forecast horizon and a length M observed TS, $\{z_\tau\}_{\tau=1}^{M}$. To make the considerations more concrete, we focus on the problem of short-term electrical load forecasting (STLF) expressing triple seasonality: yearly, weekly and daily (see

[5] for details). Our goal is to forecast the daily profile (24 hours) for the next day based on historical loads.

Data preprocessing. As the main input information, we introduce a weekly profile, which precedes the forecasted day. This profile is standardized as follows [4]: $\mathbf{x}_t = (\mathbf{z}_t^w - \overline{z}_t^w)/std(z_t^w)$, where $\mathbf{z}_t^w \in \mathbb{R}^{168}$ is the original sequence of the t-th week, $\mathbf{x}_t \in \mathbb{R}^{168}$ is its standardized pattern, and $\overline{z}_t^w$ and $std(z_t^w)$ are its mean and standard deviation, respectively. To enrich the input information, input vector $\mathbf{x}'_t$ is composed of five elements: $\mathbf{x}_t$, $\log_{10}(\overline{z}_t^w)$, which informs about the level of the TS, $\mathbf{d}_t^w \in \{0,1\}^7, \mathbf{d}_t^m \in \{0,1\}^{31}$ and $\mathbf{d}_t^y \in \{0,1\}^{52}$, which are binary one-hot vectors encoding day of the week, day of the month and week of the year.

An output pattern encodes the forecasted daily sequence as follows: $\mathbf{y}_t = (\mathbf{z}_t^d - \overline{z}_t^w)/std(z_t^w)$, where $\mathbf{y}_t \in \mathbb{R}^{24}$ is the t-th daily pattern and $\mathbf{z}_t^d \in \mathbb{R}^{24}$ is the forecasted sequence. The equation for decoding is: $\mathbf{z}_t^d = \mathbf{y}_t std(z_t^w) + \overline{z}_t^w$.

Gated recurrent cell. The proposed RNN employs the dilated RNN cell with attention (adRNNCell) introduced in [3]. It is shown in Fig. 1a) and the corresponding equations are shown in Fig. 1b). adRNNCell was designed for TS with complex seasonality. Its distinguishing features are as follows. First, it combines two dRNNCells [5] to obtain a more efficient cell, which is able to preprocess dynamically the input data. It introduces an attention for weighting the input information. The bottom dRNNCells in Fig. 1a) produces attention vector $\mathbf{m}_t$, whose components weight the inputs to the upper dRNNCell. Vector $\mathbf{m}_t$ has a dynamical character. It is adjusted to the current inputs at time t. Second, both dRNNCells are fed by two cell states $\mathbf{c}$ and two controlling states $\mathbf{h}$, i.e. recent states, $\mathbf{c}_{t-1}$, $\mathbf{h}_{t-1}$, and delayed states, $\mathbf{c}_{t-d}$, $\mathbf{h}_{t-d}$, $d > 1$. Third, the outputs of dRNNCells are split into "real output", $\mathbf{m}_t$ or $\mathbf{y}_t$, and a controlling output $\mathbf{h}_t$, which is an input for the gating mechanism in the following time steps.

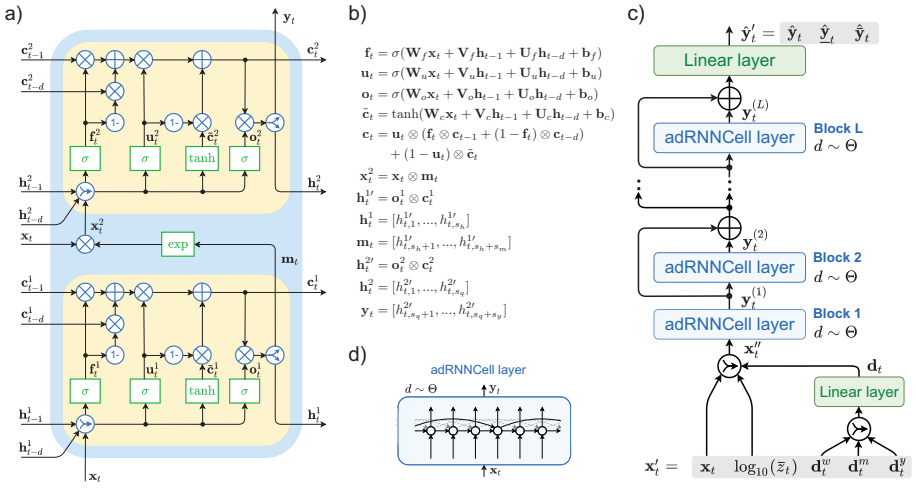


Fig. 1. Proposed RNN architecture and details.

RNN architecture. The proposed RNN extends the architecture proposed in [3] as follows. First, the number of single-layer blocks is not limited to 3. The depth of RNN is controlled by hyperparameter L. Second, the dilation of each block is selected randomly. The distribution of dilations is defined as $\Theta = \{(d_i, p_i)\}_{i=1}^{m}$, where $d_i \in \{2, 3, ...\}$ denotes a dilation, $p_i > 0$ denotes a probability ($\sum_{i=1}^{m} p_i = 1$), and m is an assumed number of possible dilations. We use an ensemble of K RNNs with dilations selected independently. Random dilations generate additional diversity among learners (random initialization is the second source of diversity).

The RNN architecture is depicted in Fig. 1c) and the adRNNCell layer is shown in Fig. 1d). The input linear layer reduces the dimensionality of the calendar variables by embedding them in r-dimensional continuous vector $\mathbf{d}_t$. The output linear layer at the top of the stacked recurrent layers, produces the point forecasts, $\hat{\mathbf{y}}_t$, and two vectors of quantiles, a lower one, $\underline{\hat{\mathbf{y}}}_t \in \mathbb{R}^{24}$, and an upper one, $\hat{\overline{\mathbf{y}}}_t \in \mathbb{R}^{24}$. These quantiles of assumed orders, $\underline{q}$ and $\overline{q}$, define the PI. RNN uses ResNet-style shortcuts between blocks to improve the learning process.

Loss function. To enable RNN to learn both point forecasts and PI quantiles, we employ the following loss function [5]:

$$L = \rho(y, \hat{y}_{q^*}) + \gamma(\rho(y, \hat{y}_{\underline{q}}) + \rho(y, \hat{y}_{\overline{q}})) \tag{1}$$

where $\rho(y, \hat{y}_q) = (y - \hat{y}_q)(q - \mathbf{1}_{(y<\hat{y}_q)})$ is a pinball loss, $q \in (0, 1)$ is a quantile order, y is an actual value (standardized), $\hat{y}_q$ is a forecasted value of q-th quantile of y, $q^* = 0.5$ corresponds to the median, $\underline{q} \in (0, q^*)$ and $\overline{q} \in (q^*, 1)$ correspond to the lower and upper bound of PI, respectively, and $\gamma \geq 0$ is a control parameter.

3 Experimental Study

In this section, we use ENTSO-E electricity demand dataset (www.entsoe.eu/data /power-stats) to verify the effectiveness of the proposed model. The dataset details the hourly loads in the period 2006-18 for 35 European countries. The RNN was optimized on data from the period 2006-17 and tested on data from 2018.

We use similar a training and optimization setup as in [4]. Based on experimentation, we assumed the dilations: $d = 2, 3, 5$ and 7 with the corresponding probabilities: $p = 0.5882, 0.1765, 0.0588$ and 0.1765. Number of blocks was $L = 3$, number of ensemble members was $K = 5$, and embedding size was $r = 6$.

Table 1 compares the forecasting quality metrics of the proposed model (ardRNNe) with the baseline models [3]. The results clearly show that ardRNNe outperforms its competitors in terms of accuracy. This was confirmed by a Giacomini-White test for conditional predictive ability. IQR(APE) shows that ardRNNe also produces the least dispersed forecasts compared to its competitors.

Fig. 2 shows the ranking of the examined models based on average MAPE for each country. Note the highest position of ardRNNe, which achieved the lowest error for 21 out of 35 countries and the second lowest error for 13 countries.

Table 1. Forecasting quality metrics.

Model	MAPE	Median(APE)	IQR(APE)	RMSE
Naive	5.08	4.84	3.32	704.34
ARIMA	3.30	3.01	3.00	475.09
ES	3.11	2.88	2.73	439.26
Prophet	4.53	4.32	3.03	619.39
FNM	2.50	2.30	2.29	334.08
GRNN	2.48	2.28	2.27	332.91
MLP	3.05	2.78	2.94	419.01
SVM	2.55	2.29	2.52	357.24
LSTM	2.76	2.57	2.52	381.76
ANFIS	3.65	3.17	3.66	507.08
MTGNN	2.99	2.74	2.69	405.18
ES-adRNNe	2.14	1.94	2.09	292.16
ardRNNe	**2.11**	**1.90**	**2.06**	**287.07**

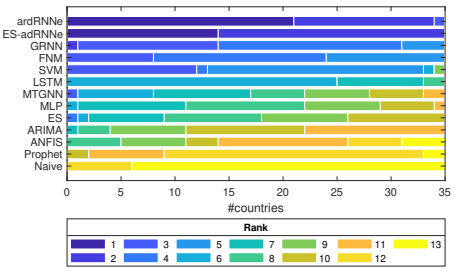


Fig. 2. MAPE ranking.

4 Conclusion

The experimental study performed on a challenging STLF problem with multiple seasonality showed that our proposed model outperforms, in terms of accuracy, its competitors including statistical, ML and hybrid models. Its superior performance is due to its hierarchical architecture, which learns temporal dependencies of different scales at different layers as well as ensembling with diversity generated by randomly dilayed connections. The model is equipped with many mechanisms and procedures designed to increase forecasting efficiency. They include: (i) new recurrent cell with dilation and attention mechanisms, which help in modeling seasonal dependencies as well as selecting input information, (ii) cross-learning, which enables RNN to capture the shared features of the individual series and prevents over-fitting, (iii) a composed asymmetrical loss function based on quantiles, which enables RNN to produce both point forecasts and PI, and also to reduce the forecast bias, and (iv) encoding the output sequences using variables determined from recent history, which helps to capture the current dynamics of the process.

References

1. Benidis, K. et al.: Neural forecasting: Introduction and literature overview. arXiv:2004.10240 (2020)
2. Hewamalage, H., Bergmeir, C., Bandara, K.: Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting. **37**, 388–427 (2021)
3. Smyl, S., Dudek, G., Pełka, P.: ES-dRNN with dynamic attention for short-term load forecasting. arXiv:2203.00937 (2022)
4. Dudek, G., Smyl, S., Pełka, P.: Recurrent neural networks for forecasting time series with multiple seasonality: A comparative study. In: 8th International Conference on Time Series and Forecasting (submitted)
5. Smyl, S., Dudek, G., Pełka, P.: ES-dRNN: A hybrid exponential smoothing and dilated recurrent neural network model for short-term load forecasting. arXiv:2112.02663 (2021)

Detection and recognition of outliers by the use of IF-THEN rules

Marcin Kacprowicz[0000−0003−4381−1359], Monika Bartczak[0000−0002−1361−3805], and Adam Niewiadomski[0000−0001−7346−5472]

Institute of Information Technology
Lodz University of Technology
Wolczanska 215, 90-924 Lodz, Poland
{adam.niewiadomski,marcin.kacprowicz}@p.lodz.pl,
monika.bartczak@dokt.p.lodz.pl

Abstract. In data mining and exploration, outliers are strange, specific, rare data in the database that cannot be ignored because they can provide potentially dangerous information. Outlier detection can contribute to detecting an illegal usage of a credit card, breaking into transaction systems at the bank, hacking into a computer system, etc. The paper presents an original fuzzy solution to the issue of outliers detection in data sets. The following novelties are introduced: definition of an outlier in terms of fuzzy logic and methods for detecting and recognizing outliers in databases. Finding outliers is done by using fuzzy rules (IF-THEN rules) based on fuzzy sets and fuzzy logic. This method is useful when linguistic knowledge rather than crisp data that are accessible. Based on the proposed new definition of the outlier and the method, the research work was conducted. Calculations and the test were based on data with the database which were expressed in a nonprecise, lingual way (similar to natural, human language).

Keywords: Outliers in databases · fuzzy rules · detection of outliers · outlying objects.

1 An outlier in terms of fuzzy rules definition

The article is a continuation of research on artificial intelligence systems using fuzzy logic. The fuzzy implications proposed by Niewiadomski and Kacprowicz [1] are applied to the general concept of outliers search presented in [2]. The application of the designed methods to the analysis of data in graph databases [3] confirmed the effectiveness of the method and allowed to formulate an outlier in terms of fuzzy rules definition. Linguistically quantfied statements based on fuzzy logic are applied to detection recognition by the authors in [4,5].
Let $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, $N \in \mathbb{N}$, be a finite non-empty set of objects. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_K\}$, $K \in \mathbb{N}$, be a set of fuzzy rules IF d_i is A_k THEN d_i is B_k, $i = 1, 2, \ldots, N$, and A_k, B_k are the antecedent and the consequent of R_k, respectively, represented by fuzzy sets in $\mathcal{D}$ (so R contains traditional IF-THEN rules). For a given $k \leq K$, *the degree of the outlier of* R_k is defined [6,7]:

$$O(R_k) = \begin{cases} \min\{\max\{T, 1-T\}, 1-C\}, & T > 0 \\ 0, & T = 0, \end{cases} \tag{1}$$

where T is *the degree of truth* (aka *conditional and unqualified proposal* [8]) evaluated as:

$$T = \frac{\sum_{i=1}^{N} \mu_{A_k \to B_k}(d_i)}{\sum_{i=1}^{N} \mu_{A_k}(d_i)}, \tag{2}$$

for $A_k \to B_k$ – a fuzzy implication, e.g. t-norm, and C – *the degree of sufficient coverage* [9], determines if the rule is activated for sufficiently large number of $d \in \mathcal{D}$ objects:

$$C = f(r_c), \tag{3}$$

where r_c is the coverage ratio:

$$r_c = \frac{1}{N} \sum_{i=1}^{N} t_i \tag{4}$$

with $t(i)$ computed as:

$$t_i = \begin{cases} 1, & \text{if } \mu_{A_K}(d_i) > 0 \text{ and } \mu_{B_K}(d_i) > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

Finally, the S-shape function $f\colon [0,1] \to [0,1]$

$$f(r) = \begin{cases} 0, & r < r_1 \\ g(r) & r_1 \le r \le r_2 \\ 1 & r > r_2 \end{cases} \tag{6}$$

where $0 \le r_1 < r_2 \le 1$ and g is a non-decreasing and continuous function on $[r_1, r_2]$.

Definition 1 (An outlier in terms of fuzzy rules). *Let $\kappa \in (0,1]$. An object $d_i \in D$, $i = 1, 2, \ldots, N$ is* an outlier *if it activates any rule R_k, $k = 1, 2, \ldots K$, such that*

$$O(R_k) \ge \kappa. \tag{7}$$

2 Example

Now, we are going to show the detection of outliers via fuzzy rules in practice. The fuzzy rules are created on the base of expert knowledge collected from bank and financial analytics, including personnel responsible for transfer security and recognition of cyberattacks on bank systems (especially phishing and unauthorized access), here, we exemplify fuzzy rules and implications as follows: let A be a fuzzy set representing linguistic label *summer* in $X = \{1, 2, \ldots, 366\}$ – numbers of days in a year in which the complaint is submitted, with μ_A given:

$$\mu_A(x) = \begin{cases} \frac{x-123}{47}, & x \in [123, 170] \\ \frac{-x+217}{47}, & x \in [170, 217] \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Let B represent label *middle county* in $Y = [5, 70]$ – per capita income in county (representing in thousands) from which come of a person who submits a complaint with $\mu_B(y)$:

$$\mu_B(y) = \begin{cases} \frac{y-35}{9}, & y \in [35, 45] \\ \frac{-y+54}{9}, & y \in [45, 54] \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Let C represent label *average time* in $Z = [0, 30]$ – numbers of days between receiving and sending the complaint to the company by CFPB (Consumer Financial Protection Bureau), with $\mu_C(z)$:

$$\mu_C(z) = \begin{cases} \frac{z-2}{4}, & z \in [2, 6] \\ \frac{-z+10}{4}, & z \in [6, 10] \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Hence the sample fuzzy rule is:

IF complaint is submitted in summer
AND submitter comes from middle county (per capita income)
THEN in average time CFPB sends complaint (11)

For example, a complaint is submitted on May 4 (the 124th day of the year) and the submitting person comes from the county where per capita income is 53 000 $. Hence, $\mu_{A_K}(124) \simeq 0.02$, $\mu_{B_K}(53) \simeq 0.11$, so via the product implication the value of the rule is $\simeq 0.0022$.
Next, for each fuzzy rule, perform calculations according to the formulas (1)-(7). A fuzzy rule for which the degree of sufficient coverage $C >= 0.1$ and the degree of outlier $O(R_k) >= 0.9$ is assumed to be an outlier. For fuzzy rule: IF complaint is submitted in summer AND submitter comes from the middle county (per capita income) THEN in average time CFPB sends complaint we received $O(R_k) = 0.91$ and $C = 0$.

3 Results and comments

The paper proposes and presents a new definition of outlier detection and recognition in databases. The definition is universal and can be applied to relational as well as non-relational datasets. A graph database was used in the study. To detect the outlier used the new definition and the IF-THEN method was used. The IF-THEN method based on different implications: product, Lukasiewicz, K_1, K_2, and K_3, and subsumed under three different defined S-shape functions. The tests were performed on 648 fuzzy rules. The application of the above implication enriched our experiment. We detected new outliers. Not all of the implications used detected, classified a given fuzzy rule as an outlier. Therefore, they were looked at in detail. It was assumed that an outlier is a fuzzy rule for which the degree of coverage $C >= 0.1$ and the degree of outliers $O(R_k) >= 0.9$. Analysis, interpretation of the results showed that the detected outliers by using other implications e.g. K2 had a high degree of uniqueness $O(R_k)$ (e.g. 0.89) for other implications e.g. PROD.

As a result, we obtained the following six unique fuzzy rules R_{85}, R_{95}, R_{121}, R_{137}, R_{149}, R_{175} (see Table 1) which are associated with 613 objects.
The definition introduced allows detection and, more importantly, recognition of specific objects (which are outliers). We detected 6 outlying rules - 613 objects. The method has been tested on real data available in the data set, where outliers may indicate incorrect data input, or even, anomalous data properties.

Table 1. Generated IF AND THEN fuzzy rules with an evaluated degree of outlier $O(R_k)$ and degree of coverage C for different implications: $PROD$, $LUKASIEWICZ$ (LUK), K_1, K_2, and K_3.

Rule No.	Fuzzy rules	$PROD$		LUK		K_1		K_2		K_3	
		$O(R_k)$	C	$O(R_k)$	C	$O(R_k)$)	C	$O(R_k$	C	$O(R_k)$	C
85.	IF complaint is submitted in the middle of spring AND submitter comes from the rich county (median household) THEN in an average time CFPB send a complaint.	0.91	0	0.92	0	0.85	0	0.98	0	0.85	0
92.	IF complaint is submitted in summer AND submitter comes from the middle county (median household) THEN in a short time CFPB send a complaint.	0.84	0	0.85	0	0.74	0	0.92	0	0.39	0
95.	IF complaint is submitted in summer AND submitter comes from the middle county (median household) THEN in an average time CFPB send a complaint.	0.93	0	0.85	0	0.74	0	0.92	0	0.39	0
121.	IF complaint is submitted in early winter AND submitter comes from the rich county (median household) THEN in an average time CFPB send a complaint.	0.95	0	0.96	0	0.88	0	0.97	0	0.9	0
137.	IF complaint is submitted in the middle of spring AND submitter comes from the rich county (per capita income) THEN in an average time CFPB send a complaint.	0.91	0	0.92	0	0.81	0	0.96	0	0.89	0
148.	IF complaint is submitted in the summer AND submitter comes from the middle county (per capita income) THEN in a short time CFPB send a complaint.	0.89	0	0.9	0	0.79	0	0.95	0	0.75	0
149.	IF the complaint is submitted in the summer AND the submitter comes from the middle county (per capita income) THEN in an average time CFPB sends a complaint.	0.91	0	0.92	0	0.9	0	0.98	0	0.87	0
175.	IF the complaint is submitted in early winter AND the submitter comes from the rich county (per capita income) THEN in an average time CFPB sends a complaint.	0.96	0	0.96	0	0.89	0	0.99	0	0.82	0

Acknowledgement This publication was completed while the second author was the Doctoral Candidate in the International Doctoral School at Lodz University of Technology, Lodz, Poland.

References

1. A.Niewiadomski and M. Kacprowicz:Type-2 Fuzzy Logic Systems in Applications: Managing Data in Selective Catalytic Reduction for Air Pollution Prevention. Journal of Artificial Intelligence and Soft Computing Research. 8597 (2021)
2. M. Kacprowicz: Search for outliers by fuzzy logic systems general concepts. TEKI(Technology, Education, Knowledge, Innovation). 6773 (2021)
3. A.Niewiadomski and M. Kacprowicz and M. Bartczak:Outliers Detection In Graph-Represented Databases Using Fuzzy Rules. Pacific Asia Conference on Information Systems, PACIS 2021 (2021).
4. A. Niewiadomski and A. Duraj and M. Bartczak: Outliers Recognition Via Linguistic Aggregation of Graph Databases. Applied Sciences, (2021), Tom 11(16), 7434, MDPI, ISSN: 2076-3417, Doi: 10.3390/app11167434, pp. 1-13,
5. A. Niewiadomski and A. Duraj: Detecting and Recognizing Outliers in Datasets via Linguistic Information and Type-2 Fuzzy Logic. International Journal of Fuzzy Systems, , nr , str. 878889. (2020).
6. B. Kosko: Fuzziness vs. probability. International Journal of General Systems (17). 11240 (1990)
7. J. van den Berg and U. Kaymak and W.-M. van den Bergh: Fuzzy classification using probability-based rule weighting. in Proc. IEEE Intl Conf. on Fuzzy Systems. 991-996 (2002)
8. G. J. Klir and B. Yuan: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Upper Saddle River, NJ:Prentice-Hall (1995)
9. D. Wu and J. M. Mendel, Linguistic summarization using IF-THEN rules. IEEE International Conference on Fuzzy Systems. 1-8 (2010)

Imbalanced data oversampling using one-class support vector machine classifier

Jakub Klikowski[0000−0002−3825−5514]

Department of Systems and Computer Networks
Wroclaw University of Science and Technology
Wroclaw, Poland
jakub.klikowski@pwr.edu.pl

Abstract. Classification of imbalanced data is an issue that still needs attention. Intelligent machine learning systems are increasingly finding various kinds of applications in the world of current technology. The major problem is insufficient data describing the minority class. Combined with the excessive amount of majority class samples, this reflects a skewed decision boundary toward the majority class. The effect of this is a relatively low predictive performance of classifiers. It is also worth noting that measuring quality itself can cause many difficulties. Inadequate selection of metrics such as accuracy can produce a seemingly good result for the highly biased model, which reflects the class distribution. One approach is data oversampling, which involves producing synthetic minority class samples. In this paper, a new technique that utilizes one-class support vector machines (OCSVM) for oversampling is proposed. Evaluated and compared with selected *state-of-the-art* methods shows promising performance and good ability to improve classification over the baseline method without oversampling.

Keywords: Imbalanced data · Oversampling · One-class SVM

1 Introduction

Classification of imbalanced data is a significant research problem [4]. Real-world data often has an uneven class distribution, which causes numerous difficulties when trying to classify it. One approach to dealing with such problems is data preprocessing [2]. These methods are split into two categories: undersampling, which reduces the majority class counts, and oversampling, which synthetically increases the number of majority class samples. One of the most recognized algorithms for oversampling is the SMOTE (Synthetic Minority Oversampling TEchnique) method [1]. The main idea of this approach is to generate new samples of the minority class sat between existing objects of this class. This method has many different variations, which create entirely new approaches based on this idea. One such variation is the Borderline SMOTE [3] approach, which primarily focuses on generating data near a potential decision boundary, where more majority class samples appear.

In this paper, a new method for oversampling imbalanced data will be proposed that uses a one-class SVM model for this purpose. A study will be conducted to verify if the proposed approach is able to improve the classification quality compared to selected *state-of-the-art* oversampling methods.

2 Proposed method

Oversampling means generating synthetic samples of a minority class to increase the importance of that class to others. These data should contain an appropriate distribution so that the relationship between objects from different classes is not disturbed. The main idea proposed in this paper focuses on selecting the best matching samples from the synthetically generated ones that follow the minority class distribution. The proposed One-Class Oversampling (OCO) method uses a one class SVM classifier for this selection process. This idea is flexible enough that other one-class models can be used.

Algorithm 1: One-Class Oversampling

Input: DS – Dataset
$OCSVM$ – One class SVM

Output: Oversampled DS

1 Divide DS into minority data (DS_m) and majority data (DS_M).
2 Train $OCSVM$ model on DS_m
3 $N \leftarrow$ Count number of samples in DS_M
4 Determine mean (μ) and standard deviation (σ) of DS_m
5 $X \leftarrow$ Generate N new synthetic samples,
where $X \sim \mathcal{N}(\mu, \sigma^2)$
6 $DS_n \leftarrow$ Predict X on $OCSVM$ model
7 $DS \leftarrow DS_M \cup DS_m \cup DS_n$
8 Return oversampled data set DS

Algorithm 1 presents the workflow of the proposed method in more detail. First, data is split into a minority class, and a majority class is performed. Then a one-class SVM classifier model is trained on the minority class data. Subsequently, the sample quantity in the majority class and the mean and standard deviation for the minority data are calculated. In the next step, new samples are generated according to the estimated normal distribution of the minority class data. Before oversampling, these samples are classified using the OCSVM model learned in step two. Any examples marked as outliers are discarded so that the minority class is reinforced with only the most matched objects. Then the newly composed data is returned by the algorithm. Project implementations are publicly available on the GitHub repository [1].

[1] https://github.com/w4k2/oco-pprai22

3 Experimental evaluation

The experimental study was conducted on a selected real-world data set with various features, samples, and imbalance ratios (Tab. 1). A 2-fold stratified cross-validation procedure with eight repeats was used for evaluation. The G-mean metric expressed classification quality. SMOTE, Borderline SMOTE (BSMOTE), Random Oversampling (ROS), and No Oversampling (NO) model algorithms were selected as reference methods. The support vector machine was used as the base classifier.

Table 1. G-mean metrics with datasets description and statistical analysis

Dataset	Features	Samples	Imb. Ratio	OCO (1)	BSMOTE (2)	SMOTE (3)	ROS (4)	NO (5)
car-vgood	7	1728	25.6	0.967 ± 0.042 *5*	0.972 ± 0.035 *5*	0.972 ± 0.037 *5*	**0.973 ± 0.032** *5*	0.783 ± 0.062
dermatology-6	35	358	16.9	0.959 ± 0.029 *2 5*	0.592 ± 0.459 *5*	0.961 ± 0.030 *1 2 5*	**0.961 ± 0.031** *1 2 5*	0.048 ± 0.128
ecoli-0-1-4-6_vs_5	7	280	13.0	**0.885 ± 0.070**	0.873 ± 0.097	0.876 ± 0.095	0.884 ± 0.096	0.875 ± 0.057
ecoli-0-1-4-7_vs_2-3-5-6	8	336	10.6	**0.851 ± 0.071** *5*	0.836 ± 0.050 *5*	0.842 ± 0.058 *5*	0.847 ± 0.052 *5*	0.745 ± 0.094
ecoli-0-1-4-7_vs_5-6	7	332	12.3	**0.889 ± 0.061** *5*	0.881 ± 0.045 *5*	0.886 ± 0.047 *4 5*	0.873 ± 0.038	0.856 ± 0.051
ecoli-0-1_vs_2-3-5	8	244	9.2	0.861 ± 0.053 *5*	0.841 ± 0.052 *5*	0.869 ± 0.044 *2 5*	**0.870 ± 0.058** *2 5*	0.812 ± 0.051
ecoli-0-1_vs_5	7	240	11.0	**0.897 ± 0.054** *5*	0.879 ± 0.061	0.888 ± 0.057	0.894 ± 0.053	0.874 ± 0.044
ecoli-0-2-3-4_vs_5	8	202	9.1	**0.893 ± 0.043** *2 5*	0.876 ± 0.053	0.892 ± 0.052 *2 5*	0.886 ± 0.051 *5*	0.867 ± 0.042
ecoli-0-2-6-7_vs_3-5	8	224	9.2	0.828 ± 0.063 *5*	0.825 ± 0.050 *5*	0.830 ± 0.058 *5*	**0.834 ± 0.055** *5*	0.789 ± 0.083
ecoli-0-3-4-6_vs_5	8	205	9.2	**0.898 ± 0.047** *5*	0.882 ± 0.063 *5*	0.887 ± 0.051 *5*	0.889 ± 0.053 *5*	0.843 ± 0.071
ecoli-0-3-4-7_vs_5-6	8	257	9.3	0.885 ± 0.038 *5*	0.884 ± 0.057 *5*	**0.889 ± 0.042** *5*	0.885 ± 0.038 *5*	0.838 ± 0.067
ecoli-0-3-4_vs_5	8	200	9.0	**0.894 ± 0.047** *5*	0.883 ± 0.066	0.888 ± 0.056 *5*	0.886 ± 0.055 *5*	0.867 ± 0.077
ecoli-0-4-6_vs_5	7	203	9.2	**0.883 ± 0.073** *2 5*	0.866 ± 0.077	0.867 ± 0.070	0.878 ± 0.066 *5*	0.853 ± 0.070
ecoli-0-6-7_vs_3-5	8	222	9.1	0.836 ± 0.065 *5*	0.825 ± 0.043 *5*	**0.839 ± 0.047** *5*	0.828 ± 0.049 *5*	0.788 ± 0.052
ecoli-0-6-7_vs_5	7	220	10.0	**0.883 ± 0.041** *4 5*	0.872 ± 0.037 *5*	0.872 ± 0.042 *5*	0.860 ± 0.036	0.838 ± 0.059
ecoli-0_vs_1	8	220	1.9	0.980 ± 0.013	0.980 ± 0.011	**0.982 ± 0.011**	0.981 ± 0.012	0.981 ± 0.013
ecoli1	8	336	3.4	0.878 ± 0.033 *5*	**0.886 ± 0.023** *3 5*	0.877 ± 0.024 *5*	0.882 ± 0.026 *5*	0.829 ± 0.058
ecoli2	8	336	5.5	**0.942 ± 0.027** *2 5*	0.924 ± 0.035	0.941 ± 0.026 *2 5*	0.937 ± 0.031 *5*	0.904 ± 0.053
ecoli3	8	336	8.6	0.890 ± 0.040 *5*	0.894 ± 0.032 *5*	0.891 ± 0.031 *5*	**0.894 ± 0.033** *5*	0.743 ± 0.066
ecoli4	8	336	15.8	**0.924 ± 0.031** *2 3 4 5*	0.897 ± 0.039	0.897 ± 0.038	0.897 ± 0.034	0.880 ± 0.079
flare-F	12	1066	23.8	**0.793 ± 0.055** *2 3 5*	0.744 ± 0.057 *5*	0.746 ± 0.057 *5*	0.775 ± 0.044 *2 3 5*	0.000 ± 0.000
led7digit-0-2-4-5-6-7-8-9_vs_1	8	443	11.0	**0.900 ± 0.015** *2 4*	0.863 ± 0.038	0.884 ± 0.032 *2 4*	0.871 ± 0.038	0.886 ± 0.037
newthyroid2	6	215	5.1	**0.925 ± 0.052** *2 3 4 5*	0.883 ± 0.047 *4 5*	0.861 ± 0.040 *4 5*	0.842 ± 0.050 *5*	0.397 ± 0.121
segment0	20	2308	6.0	0.935 ± 0.012 *5*	0.932 ± 0.014 *5*	**0.980 ± 0.006** *1 2 4 5*	0.977 ± 0.006 *1 2 5*	0.689 ± 0.025
vehicle0	19	846	3.3	0.755 ± 0.011 *5*	0.771 ± 0.015 *1 5*	**0.772 ± 0.012** *1 5*	0.772 ± 0.013 *1 5*	0.171 ± 0.213
vehicle1	19	846	2.9	**0.674 ± 0.012** *3 4 5*	0.670 ± 0.014 *3 4 5*	0.658 ± 0.017 *5*	0.660 ± 0.016 *5*	0.000 ± 0.000
vehicle2	19	846	2.9	**0.735 ± 0.029** *4 5*	0.720 ± 0.029 *5*	0.718 ± 0.024 *5*	0.712 ± 0.023 *5*	0.283 ± 0.028
vehicle3	19	846	3.0	**0.675 ± 0.018** *2 3 4 5*	0.663 ± 0.019 *4 5*	0.660 ± 0.021 *5*	0.659 ± 0.021 *5*	0.000 ± 0.000
yeast-0-2-5-6_vs_3-7-8-9	9	1004	9.1	0.793 ± 0.043 *5*	0.792 ± 0.026 *5*	**0.798 ± 0.028** *2 5*	0.796 ± 0.034 *5*	0.505 ± 0.082
yeast-0-2-5-7-9_vs_3-6-8	9	1004	9.1	**0.909 ± 0.017** *2 5*	0.901 ± 0.025 *5*	0.901 ± 0.028 *5*	0.900 ± 0.026 *5*	0.876 ± 0.028
yeast-0-5-6-7-9_vs_4	9	528	9.4	0.783 ± 0.030 *5*	0.780 ± 0.035 *5*	0.781 ± 0.036 *5*	**0.786 ± 0.032** *5*	0.134 ± 0.138
yeast-1-2-8-9_vs_7	9	947	30.6	**0.679 ± 0.090** *2 5*	0.604 ± 0.096 *5*	0.663 ± 0.067 *2 5*	0.653 ± 0.077 *5*	0.000 ± 0.000
yeast-1-4-5-8_vs_7	9	693	22.1	0.613 ± 0.057 *5*	0.575 ± 0.114 *5*	0.604 ± 0.055 *5*	**0.619 ± 0.046** *5*	0.000 ± 0.000
yeast-1_vs_7	8	459	14.3	0.698 ± 0.046 *5*	0.686 ± 0.083 *5*	**0.716 ± 0.057** *5*	0.695 ± 0.067 *5*	0.016 ± 0.063
yeast-2_vs_4	9	514	9.1	**0.896 ± 0.020** *3 4 5*	0.895 ± 0.033 *3 4 5*	0.879 ± 0.034 *5*	0.876 ± 0.030 *5*	0.750 ± 0.061
yeast3	9	1484	8.1	**0.917 ± 0.016** *2 3 5*	0.909 ± 0.012 *5*	0.911 ± 0.015 *5*	0.912 ± 0.015 *5*	0.808 ± 0.031
yeast4	9	1484	28.1	**0.820 ± 0.034** *2 3 5*	0.785 ± 0.036 *5*	0.792 ± 0.030 *5*	0.805 ± 0.017 *2 3 5*	0.000 ± 0.000
yeast5	9	1484	32.7	0.952 ± 0.022 *5*	0.954 ± 0.030 *5*	0.954 ± 0.031 *5*	**0.963 ± 0.020** *1 5*	0.291 ± 0.193
yeast6	9	1484	41.4	0.884 ± 0.036 *5*	0.871 ± 0.061 *5*	0.880 ± 0.034 *5*	**0.886 ± 0.034** *5*	0.000 ± 0.000

The results presented in Tab. 1 show the mean values with standard deviations. It is noticeable that the proposed approach obtains better results than other methods. The bold color indicates the best result among all. An important observation is that every time the proposed algorithm improves over the model without oversampling and very often obtains the best results.

Statistical analysis was also performed. Each method was compared pairwise with the other using a Student's t-test with alpha 0.05. Located below, the mean values and standard deviations indicate a statistically significant advantage of the technique in that column over the others. The information about which number defines the method is located in the first column of the table. It can be seen that the proposed approach very often obtains an advantage over the other methods. Another important observation is that OCO in the tested datasets always gets statistically significant better results than the model without oversampling.

4 Conclusions and future works

In this paper, a method for oversampling imbalanced data is proposed. Experimental evaluation has shown that the OCO method has good abilities to improve the model's predictive performance. This improvement is done at a comparable level to other state-of-the-art methods and sometimes exceeds their capabilities. Well, results are obtained for data sets with widely varying characteristics, without the influence of imbalance ratio, feature number, or samples amount.

The OCO algorithm has great potential for development. The properties of the method and different ways of generating new synthetic samples can be studied in the future. It is also worth conducting a much larger study using multi-class datasets and evaluating with more metrics.

Acknowledgements

This work was supported by the Polish National Science Centre under the grant No. 2017/27/B/ST6/01325 and by the research fund of the Department of Systems and Computer Networks, Faculty of Information and Communication Technology, Wroclaw University of Science and Technology.

References

1. Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. SMOTE: synthetic minority over-sampling technique. *Journal Of Artificial Intelligence Research.* **16** pp. 321-357 (2002)
2. Fernández, A., García, S., Galar, M., Prati, R., Krawczyk, B. & Herrera, F. Learning from imbalanced data sets. (Springer,2018)
3. Han, H., Wang, W. & Mao, B. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. *International Conference On Intelligent Computing.* pp. 878-887 (2005)
4. Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Progress In Artificial Intelligence.* **5**, 221-232 (2016)

Multi-Objective Sample Weighting for Imbalanced Data Classification

Szymon Wojciechowski[0000−0002−8437−5592]

Department of Systems and Computer Networks,
Wrocław University of Science and Technology,
Wrocław, Poland
szymon.wojciechowski@pwr.edu.pl

Abstract. An unequal sampling of classes in imbalanced datasets may lead to overfitting the model towards the majority class. Moreover, especially in probabilistic classifiers, the lack of prior probability compensation may negatively impact the classification of the minority class. Therefore, an essential element of building such a model for imbalanced data classification is sample weighting providing a more reasonable distribution estimation. However, the most commonly used approach is assigning equal weights to dense samples and the outliers, which in consequence, may lead to incorrect distribution estimation. Hence, it can be assumed that there is a more suitable sample weighting method for probabilistic classifiers. This work employs a multi-objective optimization algorithm to assign weights regarding models' sensitivity, precision, and specificity, providing a better-suited solution for imbalanced data classification. The article defines an optimization procedure for the addressed problem, evaluates the proposed method with other *state-of-the-art* methods, and outlines possible further research directions.

Keywords: Classification · Imbalanced Data · Multi-Objective Optimization

1 Introduction

The classification of imbalanced data is one of the most popular topics in machine learning community. Such data are characterized by a particular difficulty regarding the uneven representation of classes. This may result in the underrepresentation of one of them, which leads to in poorly fitted models, especially in terms of the minority class detection in such tasks as fraud detection or medical diagnosis [3]. Among the most frequently used techniques to deal with imbalanced data, one can find algorithms for oversampling the minority class, undersampling of the majority class, or a *hybrid methods* [4]. However, regardless of the chosen method, the purpose of preprocessing is to change the *prior probability* of a given problem – which in the case of some classifiers can also be achieved by proper sample weighting.

Often, weighting is based on original data imbalance, which might not be a perfect strategy in some cases. For example, giving some weight to outlier observation as the original distribution can cause a decision boundary shift. This leads to the following research question - *is it possible to propose a better procedure of assigning weights for samples in imbalanced problems*?

2 Algorithm

The proposed solution for better assignment of weights is based on a multi-objective optimization algorithm. The considered task can be defined as an optimization problem, determining a classification model characterized by the highest quality. However, evaluation of algorithms for imbalanced data classification requires specific metrics, often formulated in a way, that the error in the minority class is equally included as one in the majority class.

Commonly used *Gmean* metric is defined as geometric mean of *sensitivity* and either *specificity* or *precision*. This overcomes the limitation of single-metric optimization, which (i) is often limited to only two out of three basic metrics and (ii) depends on their aggregation. Therefore multi-criteria optimization tasks may give certain advantages in this kind of problem. Defining several metrics, which are simultaneously maximized in the optimization process, results in creating a number of interdependent solutions converging to *pareto-optimal* solutions. One of advantages is, that models can go beyond limitations of aggregated metrics, but finally, only one solution can be used, which requires a procedure for selecting most useful model.

To define given problem formally it will be assumed that the solution $s = w_1, w_2, ...w_n$ belongs to feasible solutions space S and it is defined as a series of weights w corresponding to n samples in training set. Optimization objective is to maximize three basic metrics estimated on validation set:

$$\text{maximize} \quad f_{sns}(S), \quad f_{spc}(S), \quad f_{prc}(S) \tag{1}$$

where *sns*, *spc*, *prc* translate to *sensitivity*, *specificity* and *precision* respectively. Obtained weights are further used in *Gaussian Naive Bayes* classifier training for estimating μ_y and σ_y using maximum likehood of:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{2}$$

Proposed algorithm – Multi-Objective Sample Weigthing (MOSWE) – is solving given problem using MOEAD with population size of 25 objects, limited to 10 000 evaluations. The reference directions were generated using the *Das-Dennis* algorithm [6]. To obtain a single model as a final classifier, the solution was selected using *Compromise Programming* multi-criteria decision making method.

3 Results

The experiments were conducted in stratified 5×2 CV protocol with paired t-test [1]. The proposed method was implemented in Python programming language supported by *scikit-learn* and *pymoo* packages [2]. Algorithm was compared with baseline classifier and other commonly used preprocessing techniques: *Random Undersampling* (RUS), *Cluster Centroids* (CC), *Random Oversampling* (ROS) and *Synthetic Minority Oversampling* (SMOTE) implemented in *imbalanced-learn* package [5]. For the evaluation, 21 datasets from the KEEL repository[1] were used.

Table 1. Balanced Accuracy results.

	NONE	MOSWE	RUS	CC	ROS	SMOTE
ecoli1	0.780	0.836	0.778	0.726	0.764	0.817
	4, 5	4	—	—	4	1, 4, 5
ecoli2	0.682	0.854	0.786	0.625	0.664	0.773
	4	1, 4, 5, 6	1, 4, 5	—	4	1, 4, 5
ecoli3	0.841	0.884	0.747	0.827	0.832	0.865
	—	1, 4, 5, 6	—	—	—	1, 4, 5
glass0123vs456	0.859	0.864	0.876	0.826	0.873	0.871
	4	4	4	—	1, 4	4
glass0	0.707	0.707	0.704	0.706	0.701	0.701
	—	—	—	—	—	—
glass1	0.657	0.660	0.656	0.643	0.659	0.653
	4	—	—	—	4	—
glass6	0.872	0.872	0.864	0.855	0.875	0.865
	—	4	—	—	4	—
haberman	0.575	0.559	0.608	0.561	0.621	0.600
	—	—	1, 2	—	1, 2	—
iris0	1.000	0.998	1.000	1.000	1.000	1.000
	—	—	—	—	—	—
new-thyroid1	0.987	0.979	0.973	0.948	0.977	0.978
	3, 4, 5, 6	4	4	—	4	4
new-thyroid2	0.987	0.979	0.974	0.955	0.976	0.976
	3, 4, 5, 6	4	4	—	4	4
page-blocks0	0.695	0.716	0.714	0.669	0.695	0.699
	4	1, 4, 5, 6	4	—	4	1, 4, 5
pima	0.714	0.713	0.718	0.719	0.726	0.731
	—	—	—	—	1, 2	1, 2, 3
segment0	0.896	0.914	0.890	0.860	0.887	0.892
	4, 5, 6	all	4	—	4	4
vehicle0	0.735	0.761	0.753	0.760	0.756	0.755
	—	1	1	1	1	1
vehicle1	0.677	0.697	0.670	0.627	0.671	0.675
	4	all	4	—	4	4
vehicle2	0.724	0.806	0.729	0.687	0.755	0.736
	4	all	4	—	1, 4	4
vehicle3	0.674	0.683	0.665	0.653	0.667	0.672
	—	all	—	—	—	—
wisconsin	0.966	0.965	0.965	0.964	0.966	0.966
	—	—	—	—	—	—
yeast1	0.519	0.529	0.529	0.518	0.515	0.523
	5	1, 4, 5, 6	—	5	—	1, 4, 5
yeast3	0.578	0.652	0.657	0.547	0.551	0.599
	4, 5	1, 4, 5, 6	—	—	—	1, 4, 5

Table 1 presents mean *balanced accuracy score* obtained by evaluated methods. Additionally, the list of algorithms from which referenced method was statistically better is presented below each score. It can be observed that MOSWE achieved significantly better results for the sets `segment0`, `vehicle1`, `vechicle2` and `vehicle3`, although it should be also noticed, that for the sets `ecoli2`, `ecoli3`, `page-block0`, `yeast1` and `yeast3` difference between MOSWE and RUS was not statistically significant. In the remaining datasets, algorithms did not outperformed each other, although most of the algorithms were achieving better score than CC. Interestingly, the baseline algorithm for the `new-thyroid1` and `new-thyroid2` was better then other methods, with the exception of MOSE, where there was no statistical significance. Moreover, RUS and ROS algorithms

[1] https://sci2s.ugr.es/keel/imbalanced.php

Table 2. Balanced Accuracy mean rank results.

NONE	MOSWE	RUS	CC	ROS	SMOTE
3.810	4.667	3.381	1.762	3.476	3.905

were significantly better than MOSWE only for the `haberman` dataset. Very good results of the proposed method, can be also observed in Table 2, which present the average ranking of all the methods. The observations made, allows to state that proposed method can significantly improve the results with regards to other *state-of-the-art* methods.

4 Conclusions

Conducted experiment shows that the proposed method achieves excellent results, and four datasets present the best *balanced accuracy score*. Also, there was only one dataset, for which other methods outperformed MOSWE.

It is also worth mentioning that the conducted research did not consider an important element in the classification of imbalanced data – the cost of an error made concerning the minority class is often higher than for the majority class. For example, *false-positive* test results in fraud detection are more dangerous than *true-negative*. Nevertheless, due to the pool of solutions obtained from the optimization algorithm, selecting the appropriate model may be adjusted with the expert knowledge regarding the said cost. In addition, the method is adaptable – as long as all solutions are stored, it is possible to change the model to one that corresponds to the significance of recognition of a given class set by an expert.

Acknowledgment

This work was supported by the Polish National Science Centre under the grant No. 2019/35/B/ST6/04442.

References

1. Alpaydin, E.: Combined 5 × 2 cv f test for comparing supervised classification learning algorithms. Neural Computation **11**(8), 1885–1892 (1999)
2. Blank, J., Deb, K.: Pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
3. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from Imbalanced Data Sets. Springer International Publishing, Cham (2018)
4. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**(4), 221–232 (2016)
5. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research **18**(17), 1–5 (2017)
6. Qingfu Zhang, Hui Li: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007)

Prediction of German Bundesliga football match results using ensembles of classifiers

Szymon Głowania[1][0000−0003−1137−4350]

Department of Machine Learning, University of Economics in Katowice, 1 Maja, 40-287 Katowice, Poland. szymon.glowania@uekat.pl
http://www.ue.katowice.pl/kum

Abstract. Prediction of sports performance, in particular football, is becoming an increasingly interesting and popular topic for both research and business applications. Available results are mainly based on data from English premier leagues. The current state of knowledge provides opportunities related to improvement of classification accuracy and exploration of other leagues very popular among fans and business environment (e.g. German Bundesliga).
The main objective of the paper is to verify the hypothesis whether the proposed approaches related to the use of heterogeneous ensembles of classifiers will allow obtaining predictions at a higher level of classification accuracy than classical machine learning algorithms, in particular artificial neural networks.
The proposed approach was tested on real data. Appropriate models were trained, and their accuracy was evaluated based on the results of the football matches of the German Bundesliga. The methods selected based on the literature review were experimented. The achieved results of the classification quality assessment measure confirm the realization of the set objective and hypothesis.

Keywords: heterogeneous classifiers · ensembles of classifier · sport result prediction · neural networks.

1 Prediction of sport results using machine learning

Artificial intelligence and machine learning are finding more and more applications in fields related to economy, politics, science, but also entertainment. Researchers, specialists, and enthusiasts alike are finding more and more interesting applications for the available algorithms.

The high availability of data and the growing use of sports analytics is leading to the popularization of these applications and the application of ever newer tools to them. The amount of investment by companies both in the teams or players themselves, and in parts of the sports entertainment industry, is leading to the increasing use of machine learning algorithms in this area. The use of machine learning can be found in a wide variety of sports, such as football [1, 13], basketball [11], rugby [9]. Regarding number of publications (based on to

scopus.com, as of 05.03.2022), football remains the most popular sport in which machine learning is applied. The most analyzed European league is the English Premier League [1], however, in the literature you can also find studies based on other leagues that are very popular among fans, such as the German Bundesliga [13]. This paper focuses on the analysis of the German Bundesliga due to its popularity and continuation of previous research [8].

2 Machine learning algorithms selected for experimentation

The most common approach to predicting sports performance is classification. In this approach, using data and selected algorithms, it is possible to predict the membership of a new object to one of the predefined classes. In the analyzed example, three classical classes possible for the end of the match are selected: home team win (1), visiting team win (2) and draw (0).

Classical machine learning algorithms, which are also the most popular in sports performance prediction, were used in the study. Two original approaches based on heterogeneous ensembles of classifiers are also presented to improve classification performance.

The first of the selected algorithms is decision trees. The tree presented in this paper was created based on [3, 7]. The maximum depth was set to 3, and the algorithm chosen was CART.

Another frequently used approach is the support vector machine. The linear classifier is chosen, and its implementation is consistent with [4].

Artificial neural networks are some of the most popular algorithms that are now widely used in both science and business due to the increase in computing power of computers. For the experiment, networks were created with the structure: 6–3–2–1 and 6–3–1. L-BFGS was chosen as the solver, and the maximum number of iterations is 100. The solution is implemented according to the publication [12].

Classifier ensembles are an approach based on building a classifier using a collection of individual classifiers. This approach can significantly improve the efficiency of the solution, as presented in, e.g. [6, 10].

Boosting and AdaBoost is an algorithm whose idea of operation is based on the creation of many simple classifiers, but through their number on the improvement of representativeness and consequently quality. The fathers of this approach are considered to be Y. Freund and R. E. Schapiro, and the algorithm used in this paper was implemented according to their publications [5].

Bagging is a method developed by L. Breiman to improve classification and performance of regression models due to accuracy and stability by reducing variance. The experiment uses an implementation that is consistent with the publication [2].

Random forests are a method of building an ensemble of classifiers proposed by L. Breiman. This approach is based on creating multiple decision trees and then creating a single classifier from them. A random forest with a maximum

tree depth of 3 and a number of estimators equal to 100 was chosen for the experiment, the entire operation of the algorithm is as published [3].

In this paper, two original approaches based on heterogeneous ensembles of classifiers [8] are used. The first ,,all" approach is to use previously trained classifiers: decision tree, support vector machine, AdaBoost, bagging and random forests to build a new classifier. The decision class was selected by simple voting. In the subsequent ,,selected" approach, the same algorithms were used, but the class decision had to be made unanimously by the algorithms used. This approach does not guarantee complete coverage of all decision cases, but can achieve high accuracy.

3 Conducted experiments

The necessary data set was retrieved using a solution prepared by the author. The data concerns the results of individual games and league tables in the seasons from 2011/2012 to 2020/2021 of the German Bundesliga. They cover a total of 340 matches from 10 years, resulting in 3060 records for analysis.

The experiments were conducted using the train and test method. The data was divided into a training set and a test set. The training set contained data up to and including the 18th round of the 2020/2021 season, while the test set contained data from the 19th to the 34th round of the 2020/2021 season. In accordance with previous research and the literature review, observations from the first 5 rounds of each season were removed from the set [8].

The data available about the meeting and the league table after the meeting were used in the experiment. The selected attributes are: ,,round", ,,home team's place in the league table", ,,total points scored by the home team", ,,visiting team's place in the league table", ,,total points scored by the visiting team" and ,,difference of points scored by the home team and visiting team".

The tested models were evaluated for quality using classification accuracy. The measure was calculated according to the formula $accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$ and using the confusion matrix. The results of accuracy for algorithm: Random=33,33%, Artificial neural networks 6–3–1 = 49,67%, Bagging = 49,67%, Support vector machine = 54,90%, Decision tree = 56,21%, Artificial neural networks 6–3–2–1 = 56,21%, Random forests = 56,86%, AdaBoost = **59,48%**, Approach (all) = **59,48%**, Approach (selected) = **61,54%**.

Each of the algorithms used achieved higher accuracy than the random approach. Regarding classification accuracy, the best result was obtained by the ,,Approach (selected)". In this solution, there is no guarantee of complete coverage of the test set. The next two approaches ,,Approach (all)" and ,,AdaBoost" will perform well when a decision is needed for each observation and their prediction accuracy is also at a high level.

The experiments show that the use of both classical machine learning algorithms and the proposed approaches based on heterogeneous ensembles of classifiers allows obtaining satisfactory results in terms of prediction accuracy.

4 Summary and further work

The main objective of the paper was to verify whether the proposed ,,all" and ,,selected" approaches using heterogeneous ensembles of classifiers will allow obtaining predictions at a higher level of classification accuracy than classical machine learning algorithms, in particular artificial neural networks. The goal was achieved, and the research hypothesis was confirmed by the conducted experiments. The best results in terms of classification accuracy were obtained by ,,Approach (selected)", and the most versatile in application are: ,,Approach (all)" and ,,AdaBoost".

Further work is planned to experiment with heterogeneous ensembles of classifiers by incorporating additional voting methods and extending the set with additional features.

References

1. Baboota, R., Kaur, H.: Predictive analysis and modelling football results using machine learning approach for english premier league. International Journal of Forecasting **35**(2), 741–755 (2019)
2. Breiman, L.: Bagging predictors. Machine learning **24**(2), 123–140 (1996)
3. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
4. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. the Journal of machine Learning research **9**, 1871–1874 (2008)
5. Freund, Y., Schapire, R.E., et al.: Experiments with a new boosting algorithm. In: icml. vol. 96, pp. 148–156. Citeseer (1996)
6. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence **12**(10), 993–1001 (1990)
7. Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer (2009)
8. Kozak, J., Głowania, S.: Heterogeneous ensembles of classifiers in predicting bundesliga football results. Procedia Computer Science **192**, 1573–1582 (2021)
9. McCabe, A., Trevathan, J.: Artificial intelligence in sports prediction. In: Fifth International Conference on Information Technology: New Generations (itng 2008). pp. 1194–1197. IEEE (2008)
10. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of artificial intelligence research **11**, 169–198 (1999)
11. Pai, P.F., ChangLiao, L.H., Lin, K.P.: Analyzing basketball games by a support vector machines with decision tree model. Neural Computing and Applications **28**(12), 4159–4167 (2017)
12. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986)
13. Schauberger, G., Groll, A., Tutz, G.: Modeling football results in the german bundesliga using match-specific covariates (2016)

Spatial Information in Graph Convolutional Neural Networks

Bartłomiej Wójcik and Arkadiusz Tomczyk[0000−0001−9840−6209]

Institute of Information Technology
Lodz University of Technology
al. Politechniki 8, 93-590 Lodz, Poland
arkadiusz.tomczyk@p.lodz.pl

Abstract. In recent years, to enable application of convolutional neural networks to structures other than regular grids (e.g. graphs), different definitions of convolution operation were proposed. Some of them, however, do not take into account spatial coordinates of graph nodes, if they are available in the considered dataset, and focus only on node features. In this work we propose three strategies, which allow to add that information to network architectures. The presented approach leads to results improvement on four benchmark datasets.

Keywords: Convolution · Graph neural networks · Spatial information.

1 Introduction

Convolutional neural networks CNN ([7]) are still state-of-the-art methods of image processing. Analyzing working principle of single convolutional layer[1], it is evident that such classic convolution inherently depends on spatial distribution of pixels. To be more specific, calculating result $\mathbf{h}'_i \in \mathbb{R}^m$ of single convolution layer for pixel i we consider features $\mathbf{h}_j \in \mathbb{R}^n$ of pixels j in the local neighborhood $\mathcal{N}(i)$. The relative position $\mathbf{p}_{j,i} = \mathbf{p}_j - \mathbf{p}_i \in \mathbb{R}^d$ allows to identify corresponding parameter of trainable kernel (mask).

Currently, convolutional neural networks are applied not only to regular structures (grids) but also to irregular ones (graphs). Graph nodes are also described with feature vectors, as pixels in the case of images. There are however substantial differences. First of all, the neighborhood $\mathcal{N}(i)$ is not fixed - different nodes can have different number of neighbors. Secondly, the spatial relations between nodes need not be known since in general case nodes need not to have spatial coordinates assigned (e.g. in social networks) and even if they have, those coordinates need not to be expressed by integer numbers. Finally, additional knowledge can be encoded in features of edges connecting graph nodes[2]. Consequently, the definition of convolution operation had to be modified to solve

[1] In this paper we focus only on convolutions and additional elements of CNNs achitectures like: non-linearities, normalizations, etc., are omitted.

[2] Edge features, although considered in many graph convolutional networks, are not considered in this work to focus only on spatial coordinates of nodes.

all of those problems. Most of the existing approaches ([6, 4, 8, 5, 1, 2]), can be described in the following form:

$$\mathbf{h}'_i = \mathrm{LIN}_{\mathbf{L}(i),\mathbf{l}(i)}(\mathbf{h}_i) + \square_{j \in \mathcal{N}(i)} \mathrm{LIN}_{\mathbf{M}(j,i),\mathbf{m}(j,i)}(\mathbf{h}_j) \tag{1}$$

Matrices $\mathbf{L}(i)$, $\mathbf{M}(j,i) \in \mathbb{R}^{m \times n}$ and vectors $\mathbf{l}(i)$, $\mathbf{m}(j,i) \in \mathbb{R}^m$ can be fixed or contain trainable parameters, $\square$ denotes a differentiable, permutation invariant operator, e.g. sum, mean, max, etc., whereas LIN[3] is a linear transformation of feature vectors. Some of those approaches in a natural way can handle spatial coordinates of graph nodes. The rest of them, however, do not consider it all. In this work we investigate how to use effectively the latter architectures in problems where node positions are available.

2 Method

Classic graph convolutional neural networks handle node positions in three typical ways. Firstly, some of them, e.g. MoNet [8] and SplineCNN [4], try to naturally generalize classic CNNs. It requires training of continuous kernel functions (masks) to directly operate on continous relative coordinates $\mathbf{p}_{j,i}$. They model them using either gaussian mixture model or B-spline bases, respectively. Secondly, to our best knowledge, there is one method SGCN [2] which tries to tackle the same problem without following CNN strategy. In this case:

$$\mathbf{M}(j,i) = \mathrm{diag}(\sigma(\mathbf{U}\mathbf{p}_{j,i} + \mathbf{b})) \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$ and σ is a nonlinear function (ReLU was used). A limitation of such formulation is the fact that here $m = n$. To partially avoid it, results of several operations can be concatenated. Finally, there is a numerous group of methods like: GCN [6], GAT [1], GraphSAGE [5] etc., that simply ignore those positions focusing on node features only.

The most straightforward option to take into account node coordinates $\mathbf{p}_i$ in the latter models would be adding them (concatenation) to the feature vector $\mathbf{h}_i$. This approach alone leads to the results enhancement but there is still space for further improvement. That is why in this work we propose three strategies allowing to inject additional spatial information into convolutional operation:

- $\mathbf{M}(j,i) \rightarrow \mathbf{M}(j,i) \cdot \mathbf{a}^T \mathbf{p}_{i,j}$ - linear strategy where $\mathbf{a} \in \mathbb{R}^d$ is an additional trainable parameter
- $\mathbf{M}(j,i) \rightarrow \mathbf{M}(j,i) \cdot \mathrm{MLP}_{\mathbf{A_2},\mathbf{b_2},\mathbf{A_1},\mathbf{b_1}}(\mathbf{p}_{i,j})$ - non-linear strategy where MLP[4] is a multilayer perceptron with trainable parameters $\mathbf{A_2} \in \mathbb{R}^{1 \times k}$, $\mathbf{b_2} \in \mathbb{R}$, $\mathbf{A_1} \in \mathbb{R}^{k \times d}$ and $\mathbf{b_1} \in \mathbb{R}^k$ (k denotes the number of hidden units, there is 1 output)

[3] $\mathrm{LIN}_{\mathbf{A},\mathbf{b}}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$

[4] $\mathrm{MLP}_{\mathbf{A_2},\mathbf{b_2},\mathbf{A_1},\mathbf{b_1}}(\mathbf{x}) = \mathrm{LIN}_{\mathbf{A_2},\mathbf{b_2}}(\sigma(\mathrm{LIN}_{\mathbf{A_1},\mathbf{b_1}}(\mathbf{x})))$

Table 1: Comparison of classic approaches and proposed strategies applied in GraphSAGE. All results were obtained using PyG framework (average value from 5 runs is presented). Some of them, for superpixel-based representation of MNIST available in PyG, were taken from [2] (in particular result for SGCN method).

Method	MNIST	MNIST PyG	Fashion MNIST	CIFAR10	AIDS
without position in input features					
SplineConv	97.21%	95.22% [2]	87.46%	49.53%	81.47%
GMMConv	96.84%	91.11% [2]	86.62%	45.68%	80.58%
GraphSAGE	94.15%	79.88%	84.72%	55.69%	79.73%
SGCN	-	95.95% [2]	-	-	-
Ours linear	96.72%	97.49%	87.37%	63.03%	**95.62%**
Ours non-linear	98.29%	97.01%	87.22%	65.52%	95.58%
Ours enhanced	**98.90%**	**97.91%**	**89.47%**	**67.68%**	95.09%
with position in input features					
SplineConv	97.52%	97.83%	87.50%	63.03%	95.62%
GMMConv	97.07%	96.10%	87.22%	61.80%	93.08%
GraphSAGE	98.01%	97.13%	88.09%	66.32%	93.48%
Ours linear	98.45%	97.78%	88.35%	65.11%	**97.59%**
Ours non-linear	99.05%	98.32%	88.41%	67.07%	96.96%
Ours enhanced	**99.10%**	**98.81%**	**90.98%**	**73.23%**	96.65%

– $\mathbf{M}(j,i) \rightarrow \mathbf{M}(j,i) \cdot \mathrm{diag}(\mathrm{MLP}_{\mathbf{A_2},\mathbf{b_2},\mathbf{A_1},\mathbf{b_1}}(\mathbf{p}_{i,j}))$ - enhanced non-linear strategy, here $\mathbf{A_2} \in \mathbb{R}^{n \times k}$, $\mathbf{b_2} \in \mathbb{R}^n$, $\mathbf{A_1} \in \mathbb{R}^{k \times d}$ and $\mathbf{b_1} \in \mathbb{R}^k$ (there are n outputs)

It is worth noticing that this is a generic approach. It can be used with any convolution operation that can be described by formula (1).

3 Results

In the conducted experiments we have analyzed how the proposed strategies influence the results of GraphSAGE model where $\mathbf{L}(i) = \mathbf{W}_1$, $\mathbf{M}(j,i) = \mathbf{W}_2$ are trainable matrices, $\mathbf{l}(i) = \mathbf{0}$, $\mathbf{m}(j,i) = \mathbf{0}$ and $\square$ is a mean operator. Table 1 presents the results of those experiments with coordinates $\mathbf{p}_i$ included in node feature vectors $\mathbf{h}_i$ of input signal and without them, respectively, for four different datasets. First three of them are superpixel-based representations of images (MNIST, Fashion MNIST, CIFAR10), whereas the last one (AIDS) contains graph representation of chemical molecules. To make the results comparable in all cases three convolutional layers were considered with ReLU as a non-linear activation function. Since classification task was considered convolutional layers were followed by MLP (two fully-connected layers with ReLU between them)

and while training cross-entropy lass was used. In all MLPs there were 64 hidden units.

Our results successfully demonstrate the increase of performance achieved across all four datasets. In the case of image datasets, the enhanced non-linear method was the most effective. Chemical compound AIDS dataset obtained the highest accuracy using linear strategy. Conducted experiments show also evident improvement when additionally coordinates $\mathbf{p}_i$ are included in node feature vectors $\mathbf{h}_i$. Since for the same image different superpixel-based representations can be generated, to compare results with SGCN we have performed additional experiments for MNIST dataset available in PyG framework ([3]). Also in this case our strategies lead to better results.

4 Summary

In this work we have shown that properly used spatial information can improve outcomes of graph convolutional neural networks, even if in its original form they do not consider nodes coordinates at all. We have proposed several strategies and proved (using GraphSAGE model and four well-known benchmarks) their positive influence on the classification results. Our further research will focus on experimenting with other models and datasets where spatial information is available. Moreover, we want to check other strategies as well as apply this approach for other tasks (e.g. node classification).

References

1. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks? (2021)
2. Danel, T., Spurek, P., Tabor, J., Smieja, M., Struski, L., Slowik, A., Maziarka, L.: Spatial graph convolutional networks. In: Neural information processing 27th International Conference, ICONIP 2020 : Bangkok, Thailand, November 18-22, 2020 : proceedings, part V, pp. 668–675. Communications in Computer and Information Science, ISSN 1865-0929, eISSN 1865-0937; Vol. 1333, Springer, Cham (2020)
3. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
4. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: SplineCNN: Fast geometric deep learning with continuous b-spline kernels. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 869–877 (2018)
5. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. p. 1025–1035. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017)
7. Lecun, Y., Bengio, Y.: Convolutional Networks for Images, Speech and Time Series, pp. 255–258. The MIT Press (1995)
8. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5425–5434. IEEE Computer Society, Los Alamitos, CA, USA (jul 2017)

The Impact of Using Constraints on Counterfactual Explanations

Maciej Falbogowski, Jerzy Stefanowski, Zuzanna Trafas, and Adam Wojciechowski

Poznan University of Technology, Poznań, Poland
jstefanowski@cs.put.poznan.pl

Abstract. Constraints of attributes in counterfactual explanations of machine learning models are studied. The experiments showed that adding them leads to small, acceptable decreases of the evaluation measures while ensuring the correctness of the explanations.

1 Introduction

In the last decade, intensive development of the *explainable artificial intelligence* (XAI) has been observed. It supports explanations of the predictions of machine learning models, i.e. answering the questions of why the model made such a decision for the given instance [3].

In our work we focus on *counterfactual explanations* (briefly counterfactuals). Unlike other explanation methods, they provide recommendations on how to change the description of an example to achieve the desired prediction. For instance, consider a case where someone applies for a loan and gets rejected by the system. A counterfactual explanation of this decision provides the information on minimal changes of attributes' values that will alter the decision of the system to accept this loan application instead of rejecting it. This kind of explanation is appreciated by many researchers and users, as counterfactuals are quite intuitive and may indicate to people what to do in order to achieve the desired outcome [1]. The reader is referred to [5] for a recent and comprehensive review of algorithms used for generating counterfactual explanations.

However, some limitations of these methods and their implementations could be observed. While generating a counterfactual example, most algorithms minimize certain loss functions, which may lead to changes in attribute values that are unrealistic or unacceptable for a given problem, e.g. a recommendation to change their gender or race or to reduce their age. Therefore, we share the postulates of defining additional constraints on such attributes that should be taken into account by algorithms when searching for the best explanation.

We have implemented a specialized library including new algorithms used for generating counterfactual explanations with such constraints. The aim of this paper is to experimentally evaluate the impact of using these constraints on measures of the quality of explanations for benchmark tabular datasets.

2 Counterfactual Methods and Constraints

Our library allows defining several constraints on changes of attribute values[1]. In the experiment, we explore three of the most practical ones: *freeze* – prohibits any change of the value of a sensitive attribute (e.g., gender or race of a person); *monotonicity* – indicates the preferred direction of the attribute value change (e.g., only an increase in age is allowed), *one-hot encoding* – in case of nominal attributes, changes of encoding to arbitrary real numbers are prevented and a change should preserve an appropriate zero-one encoding.

Implementations of two methods, CADEX and FIMAP, were chosen for the experiments as they were extended to address the limitations defined above. We briefly describe them below. Let $f(x) = \hat{y}$ where f is the model to be explained, x is a specific input example and $\hat{y}$ is the output of the model. Both methods aim to find a counterfactual explanation x' for which $f(x') = y'$, where $\hat{y} \neq y'$ and the attribute description of x' is as similar as possible to x.

Constrained Adversarial Examples (CADEX) [4] is a method looking for examples that change the model prediction with a minimal perturbation on attribute values. To find the best perturbations it uses the gradient descent, calculating the loss between the actual output $\hat{y}$ and the desired one y'. Following the gradient with the optimizer allows to gradually update the values of input attributes towards the decision boundary. This method is naturally suited to handle constraints by using a special mask vector for selecting attributes and limiting the direction of change.

The main idea of the Feature Importance by Minimal Adversarial Perturbation (FIMAP) method [2] is to generate minimal perturbations of the example x by a neural network g (i.e. the counterfactual of x is $x + g(x)$). As the training of g requires a differentiable model f, FIMAP uses the auxiliary neural network s which approximates predictions of the model f. The main network g and its parameters are trained in such a way that $s(x + g(x))$ should change the prediction of the model. It is achieved by optimizing the objective loss function with regularization terms by gradient descent.

3 Experiments

The constraints introduced in the previous section ensure the creation of more valid and reliable counterfactuals. For example, running CADEX for credit data without these constraints for a certain application will result in a proposal to change the nominal attribute (*Other debtors/guarantors*) from a code of 0 (*none* in one-hot encoding) to a real value (-0.459) that has no interpretation; whereas applying the constraints will result in realistic changes to these attributes.

Nevertheless the open research question is: how much the use of these constraints affects the quality of generated counterfactuals. To test this we will use four measures. For the sake of brevity, we name them by the following Greek letters:

[1] Consult https://github.com/LoGosX/CFEC for more details

- $\alpha - metric$: The distance between an example and its counterfactual [2]. The minimum difference in distance between the original example and the explanation is desired.
- $\beta - metric$: The distance between counterfactuals originated from an example and its nearest neighbor. Intuitively closest pairs of original instances should yield close pairs of counterfactuals.
- $\gamma - metric$: The fraction of attributes that were changed during the counterfactual generation process. Due to using neural networks, attributes are compared with the indiscernibility threshold of 0.1. The values of this metric are between 0 and 1. As simplicity (scarcity) of the explanation is desired, the lower value is better.
- $\delta - metric$: The binary test, which inspects if the model's decision differs between prediction for a chosen example and its counterfactual. $True = 1$ value is desired.

The experiments were conducted using two datasets from the UCI ML repository – German Credit Data (Statlog) and Adult – as they were often used in related studies and contain different types of attributes. The results are presented in Table 1.

Dataset	**Method**	**Constraints**	**Metric**			
			$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	$\boldsymbol{\gamma}$	$\boldsymbol{\delta}$
Adult	FIMAP	None	0,0201	0,0029	0,6892	0,9600
		Only One-hot	0,0930	0,0754	0,2630	0,8000
		One-hot + Monotonicity	0,1648	0,0740	0,2585	0,8750
		One-hot + Freeze	0,1724	0,0719	0,2526	0,9600
	CADEX	None	0,0180	0,0042	0,1284	1,0000
		Only One-hot	0,0497	0,0035	0,1297	1,0000
		One-hot + Monotonicity	0,0517	0,0023	0,1313	1,0000
		One-hot + Freeze	0,0495	0,0021	0,1313	1,0000
Statlog	FIMAP	None	0,0197	0,0388	0,7087	0,9300
		Only One-hot	0,1079	0,0705	0,4144	0,9400
		One-hot + Monotonicity	0,0825	0,0709	0,3921	0,4050
		One-hot + Freeze	0,0877	0,0709	0,4197	0,6750
	CADEX	None	0,0206	0,0389	0,0803	1,0000
		Only One-hot	0,0397	0,0362	0,0978	1,0000
		One-hot + Monotonicity	0,0432	0,0292	0,1188	1,0000
		One-hot + Freeze	0,0367	0,0302	0,1026	1,0000

Table 1. Evaluation metrics calculated as a mean for randomly selected 100 examples for each dataset and different constraint definitions. Monotonicity constraints were defined as increasing on *age* column for both datasets. Freeze constraints were defined on *native.country*, *sex* and *race* columns for Adult and *credit* for Statlog.

[2] Distances are calculated with HOEM metric to handle heterogeneous types of attributes. The original values of attributes are always normalized.

4 Results and Conclusions

The experimental results show that adding constraints decreases values of some metrics, but the differences are not big and they depend mainly on the method – they are negligible for CADEX while a bit more visible for FIMAP. There is no strong dependence on the type of constraint. Adding either monotonic direction or freezing an attribute led to nearly the same values of the measures in the case of CADEX. Below each metric is discussed in more detail.

The values of the α metric are generally smaller for CADEX. Defining the constraints only slightly affects the results of this method, but more strongly affects the results of FIMAP. In our opinion, it results from the sampling mechanism for discrete attributes.

The values of β metric are virtually the same for CADEX, and even slightly better with the constraints defined. For FIMAP, the values are worse but no dependence on the type of constraint is seen.

In the case of γ metric CADEX changes less attributes than FIMAP, as this method (in the default version) allows changing at most 5 attributes. It is particularly visible when constraints are not defined.

The δ metric values for CADEX are always equal to 1, since the method has access to the model and can check if its decision has been changed. In contrast, FIMAP does not have access to the original model, so it makes no such guarantee. The decrease of this metric is stronger for German data when adding constraints.
Summary: The presented experiments show that adding constraints to the methods considered does not deteriorate the explicitly considered metrics, regardless of the type of defined constraints. Declines in value are acceptable. The CADEX method achieves better evaluation values than FIMAP and, in particular, always leads to a change of the model decision into the desired value.
Ack: The research of J. Stefanowski was partially supported by the TAILOR project (EU Horizon 2020 No 952215) and 0311/SBAD/0726, while A. Wojciechowski's by AI Tech project (POPC.03.02.00-00-0001/20).

References

1. Artelt, A., Hammer, B.: Efficient computation of counterfactual explanations and counterfactual metrics of prototype-based classifiers. Neurocomputing **470**, 304–317 (2022)
2. Chapman-Rounds, M., Bhatt, U., Pazos, E., Schulz, M., Georgatzis, K.: FIMAP: feature importance by minimal adversarial perturbation. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021. pp. 11433–11441 (2021)
3. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) **51**(5), 1–42 (2018)
4. Moore, J., Hammerla, N., Watkins, C.: Explaining deep learning models with constrained adversarial examples. In: PRICAI 2019: Trends in Artificial Intelligence - 16th Pacific Rim Int. Conf. on Artificial Intelligence. pp. 43–56 (2019)
5. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: A review. arXiv: abs/2010.10596 (2020)

Universality of Forecasting Models on Water Consumption Prediction Tasks

Krzysztof Pałczyński, Tomasz Andrysiak, Magda Czyżewska,
Michał Kierul, Tomasz Kierul

Bydgoszcz University of Science and Technology
Al. prof. S. Kaliskiego 7, 85-796 Bydgoszcz, Poland
krzysztof@palczyński.com.pl

Abstract. Predicting water consumption is beneficial for water management and plays an important role in the water supply of cities. Thanks to it, the sustainable development of water resources is possible. However, in practice, water consumption is influenced by many factors and the different mechanisms of influence are complex and uncertain. Various methods for forecasting water consumption have been developed. Many methods are based on time series models that focus on past water consumption behavior and may be complemented by some exogenous variables such as a statistical regression model. The article focuses on the estimation of how much data is required in order to establish an accurate prediction of the flats water consumption without using data from these particular objects. This work examines how much data from different flats is require in order to train model in generalized fashion enabling accurate prediction of the flats unseen during training phase of the model.

Keywords: Time Series Forecasting, Neural Networks, Water Consumption.

1 Introduction

Predicting water consumption is beneficial for water management and plays an important role in the water supply of cities. Thanks to it, the sustainable development of water resources is possible. However, in practice, water consumption is influenced by many factors and the different mechanisms of influence are complex and uncertain. Various methods for forecasting water consumption have been developed. Many methods are based on time series models that focus on past water consumption behavior and may be complemented by some exogenous variables such as a statistical regression model.

The article also focuses on the estimation of how much data is required in order to establish an accurate prediction of the flats water consumption without using data from these particular objects. This work examines how much data from different flats is require in order to train model in generalized fashion enabling accurate prediction of the flats unseen during training phase of the model.

2 Datasets and Methods

In this article the data used for training and evaluation of the models comes in form of measurements of water consumption from 13 different flats from multiple estates. The measurements were conducted in the span of six months, from January 2020 to June 2020 on flats in Warsaw by SoftBlue S.A company. In order to determine both impact of selected network architecture and amount of data from different flats on results during testing, evaluation procedure has been established:

- Select model algorithm and number of flats involved during training
- Randomly select flats and put them into training set
- Take rest of flats and put them into test set
- Train the model on the test set
- Evaluate model performance on the test set

The experiments were conducted 10 times for each unique pair of model algorithm type and number of flats selected for training. The results were averaged and presented in tab. 1 and on the fig 1. ANOVA analysis and Tukey's post-hoc tests were performed in order to determine significance of differences between averaged results for each pair.

The measurements of water consumption were taken by water meters designed to send electric impulse each time 10 liters of water has flown through the device. The data was pre-processed in order to express flat's hourly water usage. The forecasts were conducted using 24-hour window. As a result, each model performed a forecast of the water consumption in the next hour based on its usage in the last 24 hours.

During nighttime it is expected that water consumption subsides. For some flats there are certain hours when water is not taken from the network at all. Because of that there are observed values of water consumption equal to 0. This is a problem for using relative metrics like Mean Absolute Percentage Error (MAPE) due to its numeric properties. The MAPE metric is calculated by

$$MAPE(Y,\hat{Y}) = \frac{1}{|Y|}\sum_{i=0}^{|Y|}\frac{|\hat{Y}_i - Y_i|}{Y_i}. \quad (1)$$

If observed value at time $t = 0$,

$$\lim_{y\to 0^+} MAPE(y,\hat{y}) = \lim_{y\to 0^+}\frac{\hat{y}-y}{y} = \hat{y}\cdot\lim_{y\to 0^+}\frac{1}{y} = \hat{y}\cdot\infty. \quad (2)$$

If predicted value is greater than zero, then MAPE error is equal to the infinity. However, if predicted value is equal to zero, then equation (2) takes form of zero times the infinity, which is an indeterminant form. Due to this limitation the metric chosen for model evaluation was Mean Squared Error (MSE) over Mean (MSEoM) given by the equation (3).

$$MSEoM(Y,\hat{Y}) = \frac{\frac{1}{|Y|}\sum_{i=0}^{|Y|}(\hat{Y}_i - Y_i)^2}{\frac{1}{|Y|}\sum_{j=0}^{|Y|}Y_j} = \frac{\sum_{i=0}^{|Y|}(\hat{Y}_i - Y_i)^2}{\sum_{j=0}^{|Y|}Y_j} \quad (3)$$

The Mean Squared Error over Mean metric combines both relative character, due to its comparison to the mean, is sensitive to significant differences between observed and predicted values and does not suffer from the numerical issues associated with division by numbers close to zero.

The models tested in this article are linear neural networks, recurrent neural networks and machine learning algorithms based on ensemble of decision trees. The linear networks are made of fully connected layers with ReLU [1] activation function. The evaluated models are one-layer network (FC-1) and two-layer network (FC-2). The recurrent neural networks are made of 2 recurrent layers and one fully connected layer. Every layer uses ReLU activation function. The evaluated recurrent models are Recurrent Neural Network (RNN) [2], Long-Short Term Memory Network (LSTM) [2] and Gated Recurrent Network [3] (GRU). The ensemble algorithms are Random Forest and XGBoost.

3 Experimental Results

The averaged results of experiments are presented in the tab. 1 and on the fig. 1. The most significant change in error rate in regard to amount of flats used during training was observed between using singular flat and two flats.

The ANOVA test has been performed for each type of network on the results from experiments clustered by the number of flats used during training. The F metric achieved for each network varies from 10.0 for FC-1 network to 25.23 for XGBoost model. The Tukey's post-hoc tests were performed in order to determine what number of flat produce error distribution separable from others.

Table 1. MSEoM for different models using varying number of flats during training phase

Flats	Models						
	FC-1	FC-2	GRU	LSTM	RNN	Random Forest	XGBoost
1	3.30%	3.60%	3.86%	3.50%	3.91%	3.35%	3.66%
2	2.95%	2.89%	3.05%	2.94%	3.06%	3.02%	3.28%
3	2.96%	2.97%	2.88%	2.80%	2.95%	2.75%	3.18%
4	2.90%	2.75%	2.88%	2.80%	2.95%	2.75%	3.07%
5	2.87%	2.73%	2.78%	2.68%	3.07%	2.69%	2.85%
6	2.90%	2.73%	2.77%	2.80%	2.87%	2.62%	2.87%
7	2.92%	2.65%	2.74%	2.66%	2.85%	2.67%	2.84%
8	2.94%	2.88%	2.71%	2.65%	2.86%	2.61%	2.78%
9	2.95%	2.63%	2.78%	2.72%	2.64%	2.59%	2.72%
10	2.94%	2.65%	2.77%	2.78%	2.79%	2.56%	2.65%
11	2.95%	2.60%	2.78%	2.68%	2.65%	2.54%	2.70%

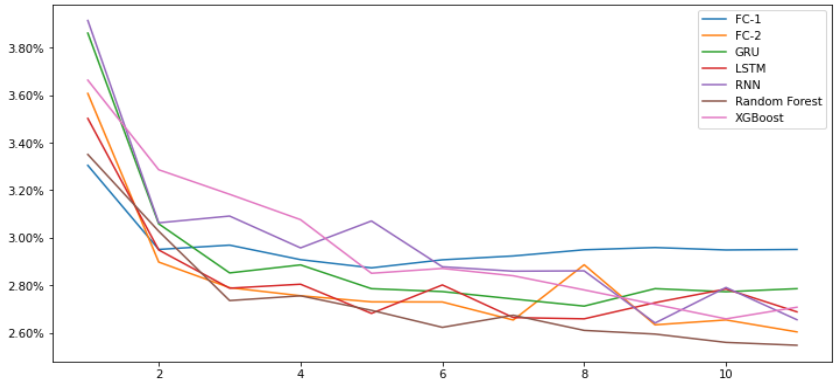


Fig. 1. Series presenting values of Mean Squared Error over Mean for different models using different number of flats for training set creation.

It turns out that error distribution of each model trained on only one flat was significantly different from error distribution of models trained on two flats. However, difference between training on two flats comparing to training on three flats was deemed not significant for every model. The only models, which results while trained on two flats differed from being trained on eleven was Random Forest and XGBoost. For every model and every pair of number of flats selected from 3 to 11 resulted in not significant difference.

The conclusion from these tests is that the smallest number of flats used during training that has significant impact on the model performance is two.

The ANOVA test has been performed on error distributions of each model trained on two flats in order to determine significance of choosing machine learning algorithm. The F metric was equal to 3.6. The Tukey's post-hoc tests were performed. It turns out that the only significant difference in error distributions were determined for XGBoost compared to FC-1, FC-2 and LSTM networks. No other model was deemed significantly different from another.

4 Conclusions

Tukey's post-hoc tests results indicates that using data acquired from two flats is enough to properly represent the task of forecasting water consumption in flats of the similar population. Although averaged error rate was smaller with each added flat for every type of model except of FC-1 and GRU, the mean differences were deemed insignificant. There was no measurable benefit of adding data from more flats.

The analysis of significance of model type selection proved that the only meaningful choice for training the model on data gathered from two flats is to not choose XGBoost algorithm. The performance of the rest of the algorithms did not prove significantly different from themselves. This notion implies selection of the least compu-

tationally expensive algorithm is preferable. However, due to suspicious, rising trend of averaged evaluation results obtained for FC-1 network, the selection of this network is not recommended despite Tukey's post-hoc test negative results. All things considered the FC-2 network is deemed the optimal model for this task.

References

1. Agarap A. F.: Deep learning using rectified linear units (relu). ArXiv Preprint ArXiv:1803.08375 (2018).
2. Sherstinsky A.: Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena 404 (132306) (2020).
3. Chung J., Gulcehre C., Cho K., Bengio Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (2014).

3D Object Localisation With 2D CNN Object Detector And 2D Odometry*

Rafal Staszak[0000−0002−5235−4201] and Dominik Belter[0000−0003−3002−9747]

Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poznan, Poland
rafal.p.staszak@doctorate.put.poznan.pl

Abstract. In this paper, we deal with the problem of objects detection and 3D position estimation by a mobile-manipulating robot equipped with an RGB-D camera and 2D laser scanner. Instead of estimating 3D position from a single image using CNN, we propose an application of CNN-based 2D object detection and gradient-based optimization that allows estimating 3D object poses from a sequence of images and robot poses obtained from an on-board 2D localization system.

Keywords: Object detection · pose estimation · mapping.

1 Introduction

An autonomous mobile robot equipped with vision, mapping, and odometry systems provides data that can be used to extract object features in an environment. In robotics applications, it is crucial to be able to determine the class and location of objects in the three-dimensional space. This might aid further execution of different robotic tasks, in which approximate object location in 3D space are needed.

In this paper, we deal with the problem of estimating the 3D position of the objects with respect to the 3D map created by the robot moving in this environment. We assume that the robot is equipped with an RGB-D camera used to register a 3D model of the environment and detect objects. Also, the robot localizes itself using onboard sensors like 2D laser scanners.

The work has the following key contribution:

- determining 3D object positions from a set of 2D detections,
- GPU implementation of a gradient-based optimisation method for estimating 3D object position.

2 Related Work

The majority of modern learning-based methods make use of deep neural networks, which both detect and predict object poses in a single-shot fashion [1–3].

* This work was parialy supported by the National Centre for Research and Development (NCBR) through project LIDER/33/0176/L-8/16/NCBR/2017 and PUT project SBAD

Such approaches excel at determining the poses in terms of translation and rotation. Moreover, they only need to be fed with a single set of image data, which speeds up the process of detecting and inferring object locations. However, the training phase of these neural networks requires training dataset containing detailed information about every object instance, its pose in space, and the bounding box in the image.

Another methods employ online optimization or matching in order to track the objects [4, 5]. These methods compute metrics similarity between the object model and the intensity of pixels in registered images. These solutions might prove to be efficient, although they require storing exact representations of either CAD models or image patterns with assigned poses. In contrast to methods that learn the 3D pose of the object and classical methods that are based on optimization only. We propose a method that utilises 2D CNN and image-based object detector [6], the estimated pose of the robot, and the efficient optimization on the GPU to estimate the pose of the detected objects in the 3D space.

3 Object Position Estimation System

During the environment scanning procedure, the object detector [6] continuously processes registered images and the information about bounding boxes, object classes, odometry measurements. After that, the detected bounding boxes are used to calculate the object middle points in every registered image. Given the odometry measurements, a set of lines $\vec{n}_i^c$ is drawn crossing the camera position m_i^c and the 3D projection of the determined middle points. In such a way, every detected object instance has a corresponding set of lines that represent the directions of those objects being observed from various viewpoints.

In a best-case scenario, the lines should intersect in the same point in space. However, the inaccuracy of the object detector parameters leads to a problem of minimising a metric error relative to the line set.

The goal of the optimization process is to estimate the position of an object of c class which is denoted as o^c. It is worth noting that the point o^c is optimized against a whole set of lines inferred from object detections for consecutively registered frames in the scanning procedure. The number of detected lines for a given class c depends on the number of input images and successful detections within a determined IoU threshold.

$$\min_o d\left(o^c\right) = \sum_{i=1}^{n} \frac{\left\|\left(o^c - m_i^c\right) \times \vec{n}_i^c\right\|}{\left\|\vec{n}_i^c\right\|}. \tag{1}$$

The minimized value is an accumulated distance between object point o^c and the whole set of corresponding lines l_i^c. This can be calculated given the anchor points m_i^c and directions of lines $\vec{n_i^c}$:

In theory, the cost function displays convex characteristics and converges quickly towards a sub-optimal minimum. If an object is observed from various viewpoints, then the resulting lines may form a conical structure. The calculated

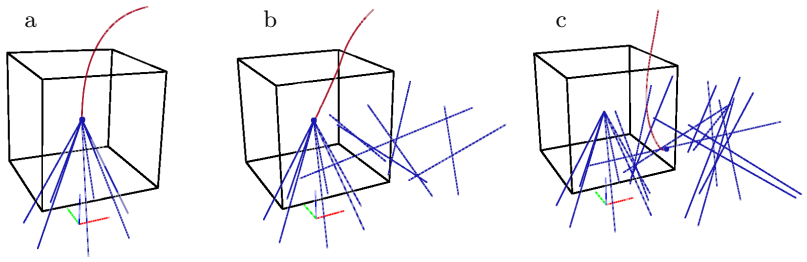


Fig. 1. Experimental estimations with various disturbance level. The intersection coordinates of 7 lines are $(0, 0, 1)$ and the unit cube showcases the scale. The estimation results in (a) and (b), which contain 0 and 7 disturbance lines, respectively, are close to the intersection point ($d_a = 0.003$) and ($d_b = 0.004$). The example with a significant disturbance (20 randomly generated lines) shows a relevant disparity ($d_c = 0.842$).

gradient is attracted towards an approximated apex of the cone. A large number of false-positive detections disturb the final outcome. However, single outliers can be outweighed by the correct detection if the number is significantly higher.

3.1 Parallel Computing

The gradient-based optimization method was formulated through matrix equations, where line parameters have been included in different components. Therefore, it was possible to make use of enormous performance in processing matrix operations and gradient calculations included within PyTorch.

4 Experiments

4.1 Datasets

Firstly, it was essential to train the object detector in order to recognise objects and extract their corresponding bounding boxes from the recorded frames. We have selected a range of objects: ring, sponge, dispenser button, probe, button, and plastic stands of 2 types. In total, 1347 sample images have been registered using Kinect for Xbox One depth sensor. Each training sample with contains annotated ground truth bounding boxes and object classes.

4.2 Results

During the experimental verification of the proposed method, the robot was continuously registering the current information about its kinematic structure, odometry, and RGBD data from the depth sensor in the given environment. The robot was controlled manually during this procedure and localized using the Slamtec Mapper M1M1 scanner. The obtained model of the environment and 3D position of the detected objects are presented in Fig. 2.

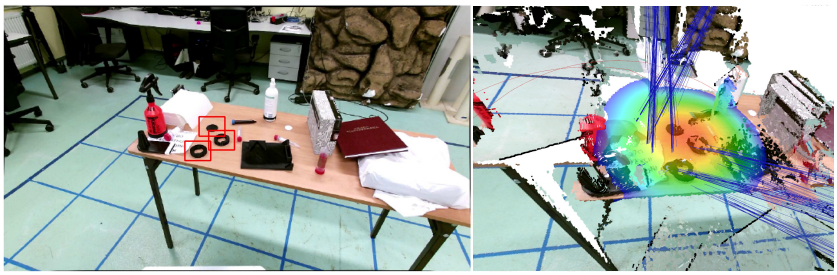


Fig. 2. Estimated positions of the objects (black rings) on the map obtained from the RGB-D camera (right). Blue lines represent measurements defined by the robot pose and the 2D detection results on the RGB images. The heat map represents proximity to the point that is found in an optimization process.

5 Conclusions And Future Work

In this paper, we present a system that estimates the positions of objects in 3D space using a 2D CNN-based detector and poses of the robot estimated by 2D lidar-based SLAM. The problem is formulated as an optimization problem that is efficiently solved using parallelized GPU implementation. In the feature, we are going to extend the problem to simultaneously estimation of objects in 3D space and robot poses like in object-based SLAM [7].

References

1. W. Kehl, F. Manhardt, F. Tombariand S. Ilic, and N. Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538. IEEE, 2017.
2. Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems*, 2018.
3. C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *Computer Vision and Pattern Recognition*, 2019.
4. T. Hodan, X. Zabulis, M. Lourakis, S. Obdrzalek, and J. Matas. Detection and Fine 3D Pose Estimation of Textureless Objects in RGB-D Images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4421–4428. IEEE, 2015.
5. Y. Konishi, K. Hattori, and M. Hashimoto. Real-Time 6D Object Pose Estimation on CPU. In *arXiv*, 2018.
6. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C Berg. SSD: Single shot multibox detector, European Conference on Computer Vision. In Leibe B. et al., editor, *LNCS vol. 9905*, pages 21–37. Springer, 2016.
7. Kyel Ok, Katherine Liu, Kris Frey, Jonathan P. How, and Nicholas Roy. Robust object-based slam for high-speed autonomous navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 669–675, 2019.

A Glimpse into the Adaptive Path Planner for a UAV

Mateusz Kosior[1][0000-0003-4368-5364]

[1] Silesian University of Technology, Konarskiego 18A, 44-100 Gliwice, Poland
mateusz.kosior@polsl.pl

Abstract. Path planning is a crucial problem in Unmanned Aerial Systems. It becomes even more complicated if more complex mission scenarios are involved, thus requiring sophisticated multi-criteria optimization algorithms. Finding a valid path in a 3D environment usable for real-time control application is a compromise between optimality and computation time. This article responds to this issue by proposing a two-level approach. First, the optimized guidance points are computed offline. Second, they are connected sequentially with fast tree-based algorithms during the flight to provide the final path for the Unmanned Aerial Vehicle. This way soft real-time capability is maintained while keeping more intricate multi-criteria optimization required to comply with the mission scenario. The article describes the general concept of Adaptive Path Planner and presents it in the context of a measurement mission employing a High-Altitude Long Endurance UAV. The idea is then supported with preliminary simulation results. Conclusions and future remarks finalize the paper.

Keywords: path planning, obstacle avoidance, unmanned aerial vehicles, UAVs.

1 Introduction

Although their development began in the 19th century, Unmanned Aerial Vehicles (UAVs) have become increasingly popular in recent years [1]. Nowadays, UAVs are considered one of the most challenging and high-potential technologies in aeronautics [2]. The number of papers related to UAVs published yearly is growing since 2008 with a more dramatic increase since 2017 [2].

One of the major problems in Unmanned Aerial Systems (UASs) is path planning [1]. It means generating an admissible flight path between points in space (waypoints) with minimal comprehensive cost [2]. And to be admissible, the path must be subject to kinodynamic constraints of the UAV. Yang et al. argues that simple 2D planning algorithms are unable to handle complex 3D environments, thus dedicated 3D algorithms are needed [1]. Nevertheless, finding a 3D complete path is classified as an NP-hard problem [1], so it is always a struggle between optimality and performance.

This paper addresses the path planning of a meteorological High-Altitude Long Endurance (HALE) UAV by proposing Adaptive Path Planner (APP). APP mixes existing general optimization algorithms with soft real-time capable algorithms used for obstacle avoidance and adaptive control.

2 Adaptive Path Planner

APP consists of two complementary modules: (1) Global Path Planner (GPP) and (2) Local Path Planner (LPP). Fig. 1 shows the methodology involving APP on an example of a general measurement mission scenario with a HALE UAV.

GPP optimizes the global path of the UAV using scenario-dependent criteria and an environment map. The map includes weather forecast, static terrain obstacles, airspace structure enforced by law and a measurement map. GPP produces global waypoints, which are scenario-optimized guiding points used as inputs to lower-level LPP. The order of following the global waypoints is strict. However, as the actual conditions may differ from predicted, the exact way of reaching global waypoints is resolved dynamically by LPP. The result is then verified in Model-In-the-Loop simulation of the flight. GPP employs biologically inspired optimization algorithms such as I-GWO, ACO, GA and PSO [1] and is used offline in Ground Control Station (GCS) for static planning.

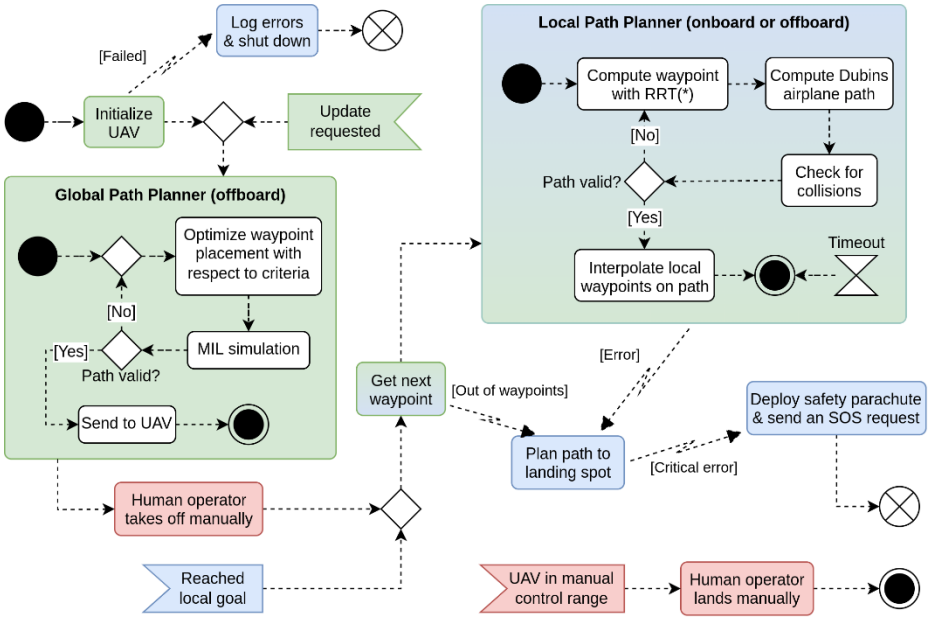


Fig. 1. A typical HALE UAV measurement mission using APP

LPP dynamically generates local paths between the global waypoints supplied by GPP. Thus, it allows the UAV to dynamically adapt to any unpredictable position errors caused by wind gusts etc. LPP connects two global waypoints with an admissible path, that is subject to kinematic constraints of the aircraft. First, it employs Rapidly exploring Random Tree (RRT) or RRT* algorithms [3] to generate a local goal waypoint. RRT and RRT* achieve soft real-time performance [1], but only grant probabilistic optimality [4]. Then, a Dubins airplane model [5] is used to validate the path by checking for collisions between the start and goal. Finally, the path is interpolated into local

waypoints used by the controller of the UAV. LPP runs online in GCS (if the UAV is in communication range) or onboard using the UAV's embedded computer (if out of range or attempting an emergency maneuver). LPP considers only static obstacles.

The mission starts in GCS by initializing the UAV and manually planning the mission scenario. Then, GPP computes the optimal global waypoints, which are then sent to the UAV. GPP can dynamically recalculate the path on demand during the mission.

Now, a human operator takes off using remote control (RC) and flies the UAV to the first global waypoint. Then, the autonomous UAS takes over the aircraft and uses LPP to plan the local path to the next global waypoint, pair-by-pair. The path is recomputed dynamically if the UAV deviates too much from the previous one. This behavior loops until all the global waypoints are reached or an error occurs. In any case, the UAS plans the path to a predefined landing spot. When the UAV is back in RC range, the human operator manually lands the aircraft. The mission is completed. If a critical error occurs during the flight to the landing spot, the UAV enters emergency mode, deploys a parachute and sends an SOS request, aborting the mission.

3 Preliminary Results

APP was implemented in MATLAB R2021b. GPP featured I-GWO, while LPP used RRT. For preliminary tests a built-in probabilistic 3D occupancy map based on OctoMap [6] was employed instead of a dedicated environment map mentioned in section 2. A sample output produced by APP for a fixed-wing UAV is shown in Fig. 2.

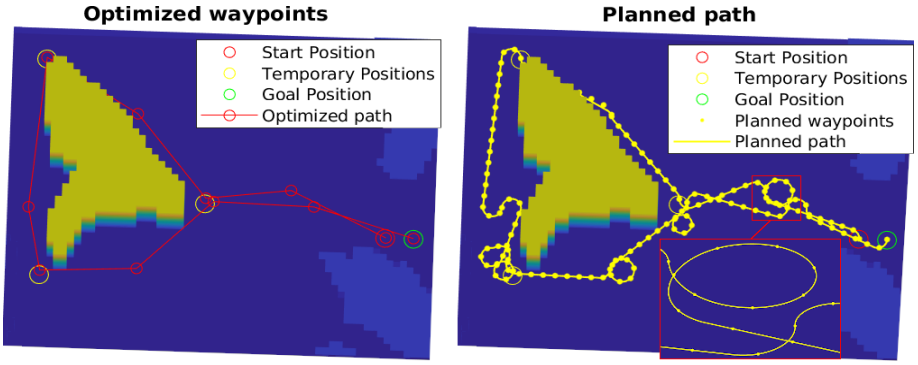


Fig. 2. Sample output of APP: global waypoints (left) and concatenated local path (right)

Fig. 2 illustrates an obstacle map of the city of Żywiec (Poland) and Żar Airport (ICAO: EPZR). The city is considered a no-fly zone and modeled as an obstacle. Red and green circles denote the points, where the control over the UAV is passed to/from the autonomous system, respectively. They should not be confused with exact takeoff/landing spots. Yellow circles denote the predefined waypoints specified by a human operator according to the mission requirements. In this example they restrict only the horizontal positions, i.e., the planner optimizes the final altitude at these points.

The left figure displays an optimized path generated by GPP. Global waypoints are represented by small red circles connected by line for clarity. Here, GPP optimizes the positions of global waypoints located between the predefined waypoints. This approach shapes the path accordingly to a set of test criteria using I-GWO.

The right figure shows the local path computed by LPP. The path concatenates several local path segments computed between the pairs of global waypoints using RRT. This example assumes the UAV has not deviated from its original path. Hence, the goal of the previous segment is the input for the next one. The “loops” seen on the figure are helices used to descend or ascend the UAV. LPP returns a set of local waypoints.

4 Conclusions

APP merges a higher-level optimization with a faster and adaptive lower-level obstacle avoidance. This keeps the optimality of the global path, which can be computed offline, while achieving soft real-time performance required for adaptive flight control.

The concept was positively verified in preliminary simulation tests in MATLAB. However, to be considered feasible, more extensive tests must be conducted to provide quantitative results. Further research will focus on the verification of GPP and LPP using model-based verification. This will involve kinematic and dynamic models of the aircraft, as well as the real HALE UAV platform.

Acknowledgments. This work is carried out under the project: "Long-endurance UAV for collecting air quality data with high spatial and temporal resolutions". It is implemented under Programme “Applied Research” under Norwegian Financial Mechanisms 2014 – 2021.

References

1. Yang, L., Qi, J., Xiao, J., Yong, X.: A Literature Review of UAV 3D Path Planning. In: Proceeding of the 11th World Congress on Intelligent Control and Automation, pp. 2376–2381. IEEE, Shenyang, China (2014).
2. Zhao, Y., Zheng, Z., Liu, Y.: Survey on computational-intelligence-based UAV path planning. Knowledge-Based Systems 158, 54–64 (2018).
3. Karaman, S., Frazzoli, E.: Incremental Sampling-based Algorithms for Optimal Motion Planning. Robotics Science and Systems VI 104(2) (2010).
4. Gopikrishnan, S., Shravan, B. V. S., Gole, H., Barve, P., Ravikumar, L.: Path Planning Algorithms: A comparative study. Space Transportation Systems (2011).
5. Chitsaz, H., LaValle, S. M.: Time-optimal paths for a Dubins airplane. In: Proceedings of 2007 46th IEEE Conference on Decision and Control, pp. 2379-2384. IEEE, New Orleans, LA, USA (2007).
6. Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots 34(3), pp. 189-206. Springer (2013).

Accurate Camera Pose Estimation from Learned Point Features: A Case Study*

Tomasz Nowak[0000−0002−2635−7732]

Institute of Robotics and Machine Intelligence
Poznan University of Technology, ul. Piotrowo 3A, 60-965 Poznań, Poland
tomasz.nowak@doctorate.put.poznan.pl

Abstract. This paper provides a case study of using a recent deep learning based approach to keypoint detection on images in the task of camera pose estimation. The application context is assisted docking to a charging station with an electric bus using monocular vision. We examined the influence of three factors on the achieved results: the backbone network, the size of the final activation maps generated by the network, and the number of convolutional layers in the keypoint head. The proposed configurations were evaluated to find the best trade-off between pose estimation accuracy (2D translation and the yaw angle estimation error were measured) and the computational complexity. The evaluation dataset was gathered using a real bus, during different weather conditions, and the ground truth data was provided by a Differential GPS. The result presented in this paper shows that proposed by us modifications of architecture can improve the accuracy of the whole processing system.

Keywords: Camera pose estimation · keypoints · deep learning.

1 Introduction

We consider a scenario of using monocular vision to guide the driver of an electric bus while docking to a charging station. An important prerequisite for successful planning of such a maneuver is to localize the bus accurately with respect to the charger's head and its supporting pylon. The bus has only a monocular camera mounted to the roof, which has to be localized with respect to some predefined points of the charger's structure. Therefore, once the charger gets detected [2], the task reduces to estimation of the pose of the camera (2D position and the yaw angle – orientation) with respect to the detected keypoints. Unlike the direct approach to camera pose estimation [7], we proposed in [3] to solve this problem using a two-stage procedure. Firstly, the keypoints are detected by a modified Faster R-CNN [4] neural network, which is also used to detect the entire charger. Secondly, a transformation between the camera frame and the charger's coordinate frame is estimated by solving the optimization problem which minimizes reprojection error upon the known locations of the keypoints

* This work was supported by PUT internal grant 0214/SBAD/0235

on the image, the real 3-D positions of those points obtained from a 3-D model of the charger, and the calibrated camera parameters [1].

This paper provides a case study of employing the recent deep neural architecture for keypoint detection: High Resolution Network (HRNet, [5]) for estimation of the keypoints in the considered task. HRNet was proposed mainly within the application context of human body pose estimation. We use this architecture in an entirely different application, where the accuracy of keypoints location is crucial. The aim of our study is to choose the HRNet configuration that yields the most accurate keypoints, maintaining also a reasonable computation complexity of the neural model.

2 Structure of the Proposed Solution

The aim of our deep learning model is to extract the keypoints selected on the charger's head and pylon. The selected points must be located in an appropriate way to provide good conditioning for the camera pose estimation problem. Hence, we selected four corners of the charger, which are harder to detect than the fiducials used in [3], but there is no need to modify the charger's appearance. The exact locations of the selected keypoints are shown in Fig.1A.

The neural network for the detection of keypoints consists of a backbone block that extracts feature maps from the image and the keypoint head which generates heatmaps from feature maps. The investigated HRNet architecture's backbone is designed to maintain a high-resolution representation of features through the whole network. Additionally, the unbiased data processing methods described in [6] were used to provide an accurate estimation of subpixel keypoints locations. The keypoint head in the default implementation has only one convolutional layer which outputs n heatmaps, where n is the number of points to be predicted. By adding a transposed convolutional layer with stride=2 (deconv) before the

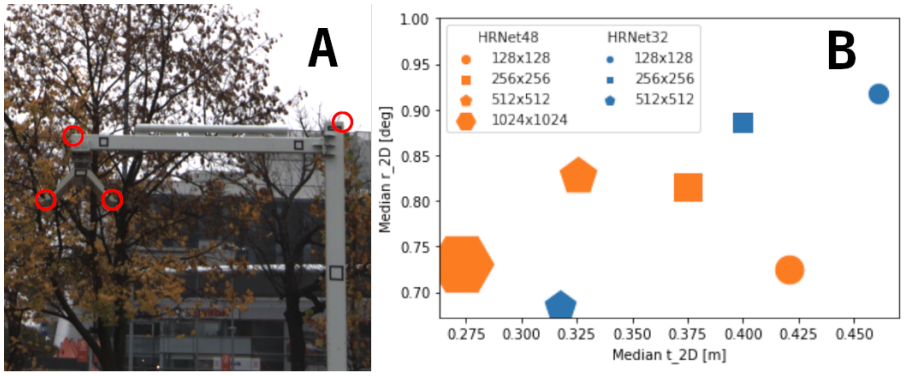


Fig. 1. Keypoints used for pose estimation (A). Translation and rotation errors for HRNet32 and HRNet48 with different heatmap size configurations. The size of marker corresponds to the number of operations required to inference single image (B).

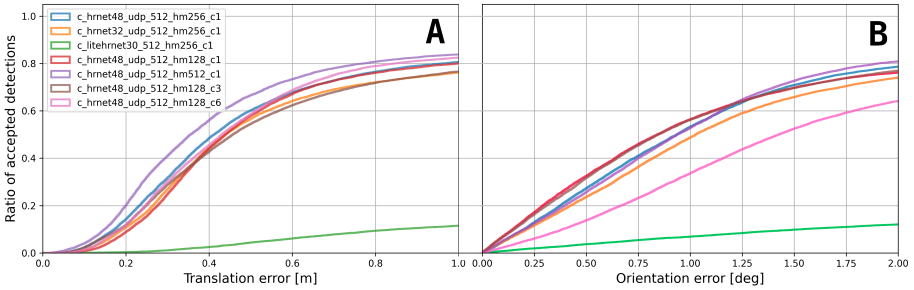


Fig. 2. Cumulative distribution functions of 2D translation error (A) and orientation error (B) for the evaluated architectures.

final convolutional layer it is possible to increase the resolution of the returned heatmap, which by default has four times smaller resolution than the input image. Extra convolutional layers followed by batch normalization and ReLU activation can be added between deconv layers and the final convolutional one to enhance the network's ability to accurately estimate the location of keypoints.

3 Experiments

The purpose of the presented experiments was to examine the influence of the neural network architecture on the 2-D pose estimation accuracy and the computational complexity. All experiments were performed off-line on a custom dataset recorded using a real electric bus and charger with the ground truth poses obtained using DGPS. The dataset consists of 81 sequences gathered over 5 days. The diversity of data was achieved by different maneuver starting points, different bus trajectories, and weather conditions. To consider the detection of a keypoint as accepted, the RMSE of the 3-D point projected on the image should be less than 10 pixels. We use three metrics to compare neural network architectures in the evaluation procedure: median of the 2-D translation error, the median of the yaw angle estimation, and the percentage of accepted detections. We selected three areas for the potential improvements in the HRNet model: the selection of an appropriate backbone network for the feature extraction, the size of the returned heatmaps, and the number of the convolutional layers in the keypoint heatmap head. For all experiments, the image input size was set to 512×512 pixels.

3.1 Backbone

In the following experiments, three backbone networks from the family of the HRNet were evaluated: HRNet48, HRNet32, and LiteHRNet. For all configurations, the heatmap size was 256×256 pixels and no extra convolutional layers were added to the keypoint head. The results show that using a bigger backbone head reduces both translation and rotation error. Moreover, it improves also the network's capability to identify all keypoints on the image (Tab. 1).

Table 1. Comparison of pose estimation errors and size of the network depending on the used backbone network, heatmap size and number of covolution layers in the head. HS means heatmap size and C depicts number of convolutional layers in the head

Configuration	Parameters [M]	Operations [GFLOPs]	Median t_2D [m]	Median r_2D [deg]	Percent of accepted detections
HRNet48 HS256 C1	63.79	87.44	0.3751	0.8148	0.944 %
HRNet32 HS256 C1	28.67	43.34	0.3998	0.8865	0.926 %
LiteHRNet HS256 C1	5.05	1.93	1.3306	2.3700	0.284 %
HRNet48 HS128 C1	63.60	84.10	0.4207	0.7242	0.937 %
HRNet48 HS256 C1	63.79	87.44	0.3751	0.8148	0.944 %
HRNet48 HS512 C1	64.84	156.56	0.3253	0.8279	0.941 %
HRNet48 HS128 C1	63.60	84.10	0.4207	0.7242	0.937 %
HRNet48 HS128 C3	63.60	84.18	0.4095	0.7199	0.937 %
HRNet48 HS128 C6	63.61	84.31	0.4702	1.3081	0.942 %

3.2 Heatmap Size

The second aspect which influences the pose estimation accuracy is the size of the output heatmaps. The default implementation of the keypoint detector based on HRNet returns heatmaps which are downsampled four times compared to the input image size, so using an image of 512×512 pixels results in 128×128 pixels heatmaps. The upsampling of the heatmaps is achieved using Transposed Convolutional layers. A single transposed convolutional layer increases the width and height of the heatmap twice. The compared configurations use HRNet48 as the backbone without extra convolutional layers in the head. The influence on the percentage of accepted detections is marginal, but increasing the heatmap size significantly reduces translation error. Further increasing the size of the heatmap significantly increases computational cost and makes the network prone to overfitting (Fig. 1B and Tab. 1).

3.3 Keypoint Head Depth

For the comparison of the influence of the number of convolutional layers in the head, we used a network with HRNet48 as the backbone and 128×128 pixels heatmaps. Increasing the number of convolutional layers to 3 improves the pose estimation accuracy with respect to both translation and rotation. However, a network with 6 layers performs worse than the smaller versions. This accuracy loss is attributed to the overfitting of the network (Fig. 2 and Tab. 1).

4 Conclusion

We demonstrated that it is possible to adopt the HRNet architecture to the task of keypoints detection for camera pose estimation, finally achieving better results than in [3], in spite of the lack of fiducials on the charger. In the considered application, the HRNet48 HS512 C1 configuration is the best choice as we do not have tight constraints on the computing resources while achieving the highest possible accuracy reduces the risk of failed maneuver. Further research will concern modifications to the loss function that should introduce geometric priors stemming from the known pattern of keypoints.

References

1. Hartley, R. I. and Zisserman, A.: Multiple view geometry in computer vision, Cambridge University Press (2004).
2. Nowak, T., Nowicki, M., Cwian, K., Skrzypczyński, P.: How to improve object detection in a driver assistance system applying explainable deep learning. IEEE Intelligent Vehicles Symposium, Paris, 226–231, (2019).
3. Nowak, T., Nowicki, M., Cwian, K., Skrzypczyński, P.: Leveraging object recognition in reliable vehicle localization from monocular images. Automation 2020: Towards Industry of the Future, AISC, Vol. 1140, Springer, 195–205, (2020).
4. Ren, S., He, K., Girshick, R. Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. NeurIPS, 91–99, (2015).
5. Wang, J., Sun, K., Cheng, T., *et al.*: Deep high-resolution representation learning for visual recognition. IEEE Trans. on PAMI, 43, 3349–3364, (2021).
6. Huang, J., and Zhu, Z., Guo, F., *et al.*: The Devil is in the Details: Delving into Unbiased Data Processing for Human Pose Estimation. arXiv:2008.07139, (2020).
7. Xiang, Y., Schmidt, T., Narayanan, V. Fox, D.: PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes. Proc. Robotics: Science and Systems, Pittsburgh, (2018).

An Embedded Deep Learning Architecture for Real-Time Place Recognition from Omnidirectional Images

Marta Rostkowska

Institute of Robotics and Machine Intelligence,
Poznań University of Technology, ul Piotrowo 3A, 60-965, Poznań, Poland
martarostkowska89@gmail.com

Abstract. This paper presents a real-time solution for place recognition in indoor environments using a convolutional neural network for extracting embeddings from omnidirectional images, which allows the robot to register a description of the entire surroundings. The proposed neural network recognizes places on distorted images from a catadioptric camera, in contrast to the more widely used approach which is based on producing panoramic images from omnidirectional images, which involves many mathematical transformations. The proposed solution achieves robust place recognition results owing to efficient retrieval of embeddings created exploiting transfer learning and fine-tuning on a limited number of actual omnidirectional images. The localization system is implemented on a NVIDIA Jetson TX2 computer with a general purpose graphics processing unit. The proposed neural network architecture makes it possible to process the omnidirectional images in real-time on this embedded hardware, which provides cost and energy efficient means of appearance-based localization for indoor service robots

Keywords: Place recognition · deep learning · omnidirectional vision.

1 Introduction

One of the most important aspects of robot autonomy is the ability to determine agent's location in the environment. Passive cameras are arguably the most popular sensors for robot localization, while particularly interesting are the omnidirectional cameras that enable the whole local scene to be registered in one image. Omnidirectional images are convenient for appearance-based visual localization, called also place recognition. This approach yields information about the similarity of the places observed in the current perception and locations stored in a database [3]. Although appearance-based localization does not provide metric information about the position of the robot in a global reference system, the ability to tell if the robot is close to one of the known locations is often sufficient for indoor navigation.

Therefore, we propose a novel approach that adopts a Convolutional Neural Network (CNN) architecture to process the omnidirectional images for real-time place recognition. The proposed system exploits the concept of global image descriptors, which was already proved to be efficient in place recognition [1]. We employ a CNN to produce the descriptors directly from the omnidirectional

images, thus avoiding the computation overhead required for producing undistorted panoramic images, which are typically used in place recognition systems for catadioptric cameras [6].

2 Localization System Overview

The localization procedure is based on finding the omnidirectional image from a known database that is most similar to the one currently acquired by the robot. As the database images are registered at known locations, finding the one that has the minimal distance (in the sense of appearance similarity) to the current perception makes it possible to roughly localize the robot.

We use a Labbot mobile robot with an integrated omnidirectional vision sensor [4] placed on top (Fig. 1a). The catadioptric sensor consists of a Microsoft Life Cam camera with a hyperbolic mirror which provides a 360° field of view and yields images in 640×480 resolution. The sensor is equipped with a NVIDIA Jetson TX2 computer with an integrated 256-core Pascal architecture General Purpose Graphics Processing Unit (GPGPU). This unit is enough to run our localization system in real-time.

In this research a dataset of 606 images (Fig. 1c and 1d) describing the robot's environment was acquired in one of the Poznan University of Technology buildings (Fig. 1b). In order to remove the areas in the images that do not carry useful information, the raw images are masked, which removes the area outside the hyperbolic mirror, and the area reflecting the camera (Fig. 1e). These images are processed by our CNN to obtain embeddings of the images. Finally, descriptors of 2048×1 size are computed for each image and stored in a database of 2048×n size which is our global map for appearance-based localization over n reference images (n=484 in the experiment). Then the algorithm creates an index from the global map (using Faiss[2] library), which is used for efficient similarity search. All these operations are accomplished off-line.

The main localization task is done on the Jetson platform in real-time. First, the CNN model and the index of images are loaded to the memory, then the candidate images are being found using KNN search in the descriptor space among the descriptors of images from the database. The real-valued descriptors are compared using the L2 distance, which turned out to be more computationally efficient than binarizing the embeddings and using the Hamming distance.

3 Deep Learning Architecture

The advantage of CNN in the image description task over traditional descriptors is related to the ability of a CNN to extract rich features. The learned descriptors are more robust to changing lighting or changes in the robot orientation than classic global image descriptors, particularly, if an extensive data augmentation process is applied while learning to disregard these changing factors.

The procedure of extracting image features and storing them in an efficient format is called embedding. It makes possible to access the feature-based description without having to pass the images from a database through the same

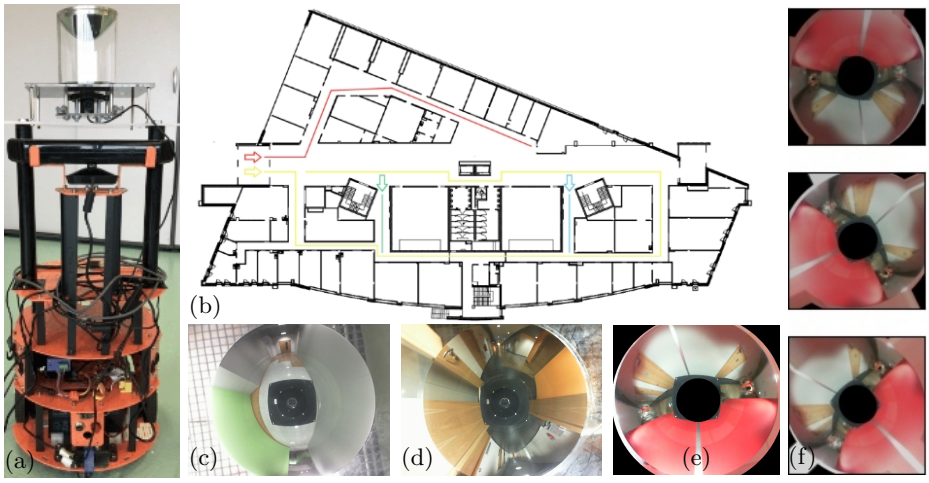


Fig. 1: Labbot robot with the catadioptric vision system (a), robot paths while collecting the images – different colors indicate different paths, then divided into segments (b), omnidirectional images of different locations (c,d), an omnidirectional image after masking (e), and examples of data augmentation (f).

neural model as the query image, as it is done, e.g. by Siamese networks [7] used for image retrieval.

We have tested a number of CNN architectures as the feature extractors in our system, finally choosing the EfficientNet [5] in B5 variant, which has 577 layers, with the input image size defined as (456,456,3). This network has high accuracy with a relatively small number of model parameters, which positively affects the processing speed in our embedded system. Due to the fact that the EfficientNet B5 was pre-trained on images (Imagenet dataset) not related to the target dataset, the network was fine-tuned before use, unfreezing a number of layers and using the categorical crossentropy loss function. This process was implemented using the dataset of around 10000 augmented omnidirectional images, produced from the previously gathered database (Fig. 1f).

A practical problem in the considered scenario was the high self-similarity of the indoor environment. As the images were acquired roughly every 0.5 m along the robot path, the neighboring images in the original database are very similar and often indistinguishable to human being. Therefore, the entire dataset was divided manually into 17 different sections, each section describing a topologically different location. Then, the localization process is executed only with respect to these 17 meaningful locations, while each of them is represented by 30 to 40 acquired images, which are partially overlapping. In the training process, each section was divided into the training (60%), validation (20%) and test (20%) sequences.

The best training results were obtained for unfrozen 50 last layers, learning rate of $1e^{-4}$ and batch size 16, with the resulting training loss: 0.1605, training accuracy: 0.9596, validation loss: 0.1183 and validation accuracy: 0.9796 (Fig. 2).

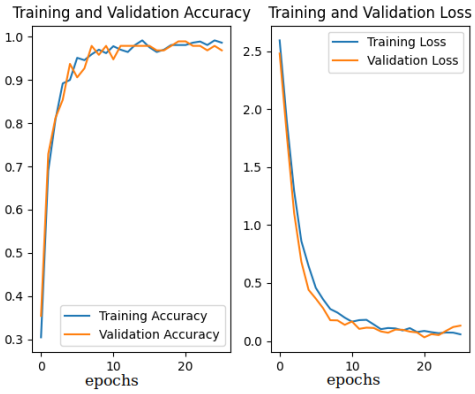


Fig. 2: Model training results

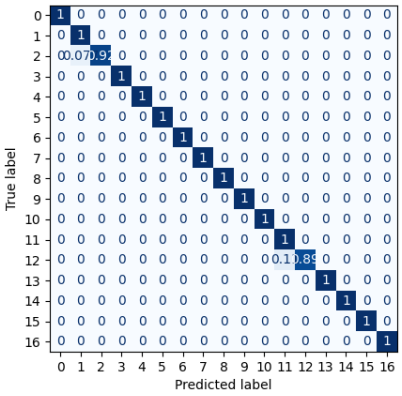


Fig. 3: Confusion matrix for 17 sections

4 Results

On the test dataset containing 122 pictures, the average accuracy of place recognition was 98% (Fig. 3), while the average processing time of a single picture was 480ms, with standard deviation of 83ms and max time of 1313ms, which allows localization at frame rate of the robot's camera. An example of correct place recognition is given in Fig. 4. The most often sections mismatching is related to a situation where the same place is the beginning of a new section and the end of the previous one. Errors are also caused by blurred images and light spots.

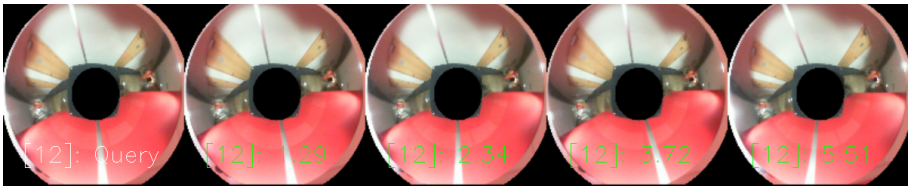


Fig. 4: Results of sample section predictions. The first image is a query, the others are the four closest neighbors. In square brackets there is the section number, and next to it is the L2 distances between the query and presented image.

5 Conclusion

This short paper demonstrated that a CNN can be trained efficiently, using transfer learning and fine-tuning approach, to produce embeddings that describe distorted omnidirectional images in an appearance-based localization system. The proposed architecture makes it feasible to run the entire process in real-time on-board of an integrated sensor with an embedded Jetson TX2 computer. Further research concerns applying spherical representations to the omnidirectional images to avoid the inactive areas, and employing a more advanced learning technique, such as triplet loss.

References

1. Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., Romera, E.: Towards life-long visual localization using an efficient matching of binary sequences from images. IEEE Int. Conf. on Robotics and Automation, (2015), 6328–6335.
2. Facebook AI Research: Faiss, https://github.com/facebookresearch/faiss. Last accessed 28 March 2022.
3. Lowry, S., Sunderhauf, N., Newman, P., Leonard, J., Cox, D., Corke, P., Milford, M.: Visual place recognition: A survey. IEEE Transactions on Robotics, 32, (2016), 1–19.
4. Rostkowska, M., Skrzypczyński, P.: Hybrid field of view vision: From biological inspirations to integrated sensor design. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI), Baden-Baden, (2016), 653–658.
5. Tan, M., Le, Q, V.: EfficientNet: Rethinking model scaling for convolutional neural networks. Proc. 36th Int. Conf. on Machine Learning, PMLR 97:6105-6114 (2019).
6. Wang, T., Huang, H., Lin, J., Hu, C., Zeng, K., Sun, M.: Omnidirectional CNN for visual place recognition and navigation. IEEE Int. Conf. on Robotics and Automation (ICRA), Brisbane, (2018), 2341—2348.
7. Zemel, R., Koch, G., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. ICML Deep Learning Workshop, Lille, France, (2015).

CNN-based Traffic Sign Detection on Embedded Devices*

Aleksandra Kos[1,2][0000−0001−9726−4472] and Karol Majek[1][0000−0002−1351−8496]

[1] Cufix, 05-825 Grodzisk Mazowiecki, Poland
[2] Poznan University of Technology, 60-965 Poznan, Poland
aleksandra.kos@doctorate.put.poznan.pl

Abstract. Traffic sign detection is a key task in autonomous driving. In addition to high accuracy, the algorithm must operate in real-time on an embedded device. Traffic signs are often found occupying a small area of a high-resolution image and can be easily confused with other signs and billboards. We analyze the aforementioned challenges, using the YOLOv4 model, which we train on the Mapillary Traffic Sign Dataset (MTSD) with a designed data augmentation method and weighted loss function. We achieve $AP_{50} = 59.0\%$ on the validation dataset. The contribution of this work is a quantized YOLOv4 traffic sign detector with an input resolution of 960×960px. The proposed model is optimized to achieve better performance on devices with limited computational resources. Our model runs at 11.2 FPS on Nvidia Jetson Xavier AGX.

Keywords: Traffic sign detection · Mapillary Traffic Sign Dataset · YOLOv4.

1 Introduction

Robotic applications of object detection algorithms often require implementation on devices with limited computational resources and memory. To run the existing object detectors in real-time, lightweight models and limited input resolution are required, which can lead to poor detector accuracy [1, 9]. To alleviate these problems, techniques for optimizing the model inference performance, such as quantization and pruning [4, 5], are used. Our goal is to analyze the challenges of real-time traffic sign detection and suggest a method that deals with this problem on embedded devices. In this work, we use MTSD to train the YOLOv4 model, then optimize our network with the tkDNN [9] library for inference on an embedded device, and assess the detection quality with 12 COCO [6] metrics. The contributions of this work are as follows: a detailed analysis of a large traffic-signs detection dataset (MTSD), a trained model capable of detecting signs belonging to 314 classes ($AP_{50} = 59.0\%$), and analysis of the results and suggestions on how to improve the system.

* The research was supported by the Ministry of Education and Science as part of the "Doktorat Wdrożeniowy" program (DWD/5/0203/2021).

2 Related Work

In recent years, many real-time object detection methods, such as [1], as well as autonomous driving datasets [11, 3] have been published. These datasets provide a variety of data from different sensors, but lack detail in road sign classes. Until recently, a thorough examination of detectors in traffic sign detection was practically impossible, due to the absence of a large dataset that would contain realistic data [2, 8, 13]. Traffic sign detection can be solved using single-stage object detection methods with large input image size, as in TSingNet [7], which achieves 20.6 FPS using the desktop Nvidia GeForce GTX 1080 GPU. Object detection networks can be optimized to achieve high performance on embedded systems when moving to embedded GPUs, such as Nvidia Jetson series models [12].

3 Dataset Analysis

In Tab. 1 we demonstrate the main features of selected traffic sign datasets. MTSD is characterized by the largest number of object instances and classes, as well as the highest variability in image and object size. It has 41909 labeled, and

Table 1. Comparison of traffic sign datasets. Sizes were calculated as geometric means $s = \sqrt{w \cdot h}$

Dataset	Images	Objects	Classes	Image size	Object size	Country	Year
RTSD [8]	59188	104358	198	1052.02±188.95	38.76±22.85	Russia	2016
TT100K [13]	16811	26349	182	2048.00±0.00	45.85±31.62	China	2016
MTSD [2]	41909	206388	314	2837.99±911.44	63.10±71.62	Global	2019

10544 unlabeled images. Each object is annotated with an axis-aligned bounding box and an identifier of one of the 314 classes. The classes are grouped into 5 main categories: *information*, *complementary*, *regulatory*, *warning* and *other*. *Other* is both a category and a class, and accounts for about 70% of objects. In addition

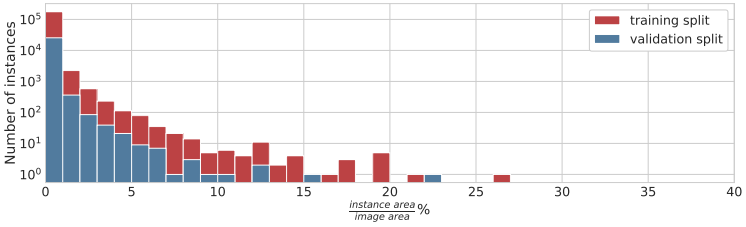


Fig. 1. Histogram of relative object areas in the MTSD dataset.

to the uneven class distribution, many images are larger than 10 MPx and most objects take up less than 1% of that area (see Fig. 1).

4 Results

We trained two YOLOv4 [1] object detectors with 960×960 input resolution and the SGD optimizer with momentum and weight decay. The first training process involved minor color and geometric transformations and allowed us to achieve $AP_{50} = 41.5\%$. The second model was trained on a dataset obtained with a more complex data augmentation method, which together with the use of the weighted cost function, increased the AP_{50} to 59.0%. The model with higher AP_{50} was optimized (FP32, FP16, INT8) using tkDNN [9] to enable inference on Nvidia Jetson Xavier AGX. In Table 2 we show the impact of quantization on the quality and speed of the detector.

Table 2. COCO metrics [6] and speed (defined as frames per second) of YOLOv4 on the MTSD validation set.

Precision	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
FP32	4.2	**34.4**	**59.0**	**35.6**	**31.9**	51.3	56.7	**44.9**	53.5	**53.6**	50.2	**64.3**	**68.5**
FP16	9.0	**34.4**	58.9	**35.6**	**31.9**	51.3	56.4	**44.9**	**53.6**	**53.6**	**50.3**	**64.3**	67.9
INT8	**11.2**	33.4	56.1	35.3	29.2	**52.7**	**58.9**	43.9	51.0	51.1	47.0	64.0	68.4

Considering the reduced input resolution and the large number of small instances, we decided to recalculate the metrics, discarding tiny ($s < 8^2$) and very tiny ($8^2 \leq s < 16^2$) objects [10]. The new validation datasets had 19 419 and

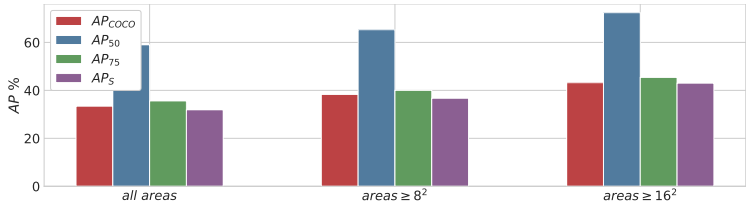


Fig. 2. The impact of discarding tiny and very-tiny objects on the average precision.

11 518 instances, respectively, compared to 26 101 objects in the original dataset. Fig. 2. shows the resulting changes in AP.

5 Conclusion and Future Work

The results presented in Fig. 2 show that the tiny and very tiny objects decrease the average precision (AP) of traffic signs detection (by up to 9.8% in our experiments). Optimizing the model does not degrade the quality, but allows for 3 times faster inference, as shown in Tab 2. For the INT8 model, an increase

in AP_M and AP_L was observed. An in-depth analysis of the effect of sample selection on quality is necessary, and we consider it to be our future work. In addition, we plan to investigate techniques such as two-step detection (utilizing Region of Interest proposals), detection in original resolution, validation on additional datasets, and analysis of AP for tiny and very tiny objects.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Ertler, C., Mislej, J., Ollmann, T., Porzi, L., Neuhold, G., Kuang, Y.: The mapillary traffic sign dataset for detection and classification on a global scale. In: European Conference on Computer Vision. pp. 68–84. Springer (2020)
3. Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A.S., Hauswald, L., Pham, V.H., Mühlegg, M., Dorn, S., et al.: A2d2: Audi autonomous driving dataset. arXiv preprint arXiv:2004.06320 (2020)
4. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
5. Liang, T., Glossner, J., Wang, L., Shi, S., Zhang, X.: Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing **461**, 370–403 (2021)
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
7. Liu, Y., Peng, J., Xue, J.H., Chen, Y., Fu, Z.H.: Tsingnet: Scale-aware and context-rich feature learning for traffic sign detection and recognition in the wild. Neurocomputing **447**, 10–22 (2021)
8. Shakhuro, V.I., Konouchine, A.: Russian traffic sign images dataset. Computer optics **40**(2), 294–300 (2016)
9. Verucchi, M., Brilli, G., Sapienza, D., Verasani, M., Arena, M., Gatti, F., Capotondi, A., Cavicchioli, R., Bertogna, M., Solieri, M.: A systematic assessment of embedded neural networks for object detection. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 937–944. IEEE (2020)
10. Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S.: Tiny object detection in aerial images. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 3791–3798. IEEE (2021)
11. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2636–2645 (2020)
12. Zhang, X., Lu, H., Hao, C., Li, J., Cheng, B., Li, Y., Rupnow, K., Xiong, J., Huang, T., Shi, H., et al.: Skynet: a hardware-efficient method for object detection and tracking on embedded systems. Proceedings of Machine Learning and Systems **2**, 216–229 (2020)
13. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2110–2118 (2016)

CNNs for State Estimation of Articulated Objects*

Kamil Młodzikowski and Dominik Belter

Institute of Robotics and Machine Intelligence, Poznan University of Technology,
60-965 Poznań, Poland
kamil.mlodzikowski@put.poznan.pl

Abstract. In this paper, we deal with the problem of state estimation of articulated objects during robotic interaction. The robot equipped with an RGB-D camera has to estimate the joint position and rotation of the articulated object when manipulating the object. The problem of accurate state estimation is challenging due to the properties of the RGB-D sensor. The known solutions require some assumptions about the shape of the objects. In this paper, we propose the application of Convolutional Neural Networks to the state estimation of articulated objects from two pairs of RGB-D images.

Keywords: articulated objects · robot perception · deep learning in robotics

1 Introduction

Mobile-manipulating robots working as personal assistants, that help with daily household tasks, should be capable of operating in an unstructured indoor environment. Similarly, robots operating in warehouses, hospitals, or factories share their workspace with humans and should deal with objects in these environments. When interacting with articulated objects, the robots should estimate the state of the object to determine if the interaction is successful (e.g. the robot opened the door).

In this research, we focus on the problem of estimating the configuration of the rotational articulated object from a sequence of RGB-D images. An example scenario is presented in Fig. 1. We propose a system that estimates the rotation axis and the configuration of the joint (angle of rotation) during the interaction. We utilize a pair of RGB-D images from a depth camera that observes the scene with moving objects. We propose the application of a deep neural network to deal with challenges related to this task (occlusions and missing depth data during interaction). Because end-to-end learning is impossible in this case, we suggest three separate neural networks that form a cascade that solves the given problem.

* The work was supported by the National Science Centre, Poland, under research project no UMO-2019/35/D/ST6/03959.

Fig. 1. In the application scenario the robot observes the scene and, from two pairs of images, concludes about the configuration of the articulated object (blue axis).

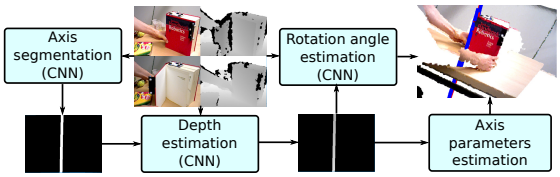


Fig. 2. Block diagram of the procedure for kinematic structure estimation of articulated objects. Each block of the system is explained in detail in the text.

1.1 Related Work

Most of the research on the interaction of robots with articulated objects is focused on visual or force/tactile perception. The RBO dataset [4] contains a set of RGB-D sequences with data from the RGB-D camera, state of the joint, and force/torque data measured during interaction with the objects. In [7] the parameters of the articulated objects, like the axis of rotation/translation or parts' poses are not estimated directly. The knowledge about articulated objects and corresponding actions can be stored in a graph-like structure [3]. The non-parametric belief propagation algorithm proposed in [2] estimates the pose of the articulated objects, but the presented approach assumes that the model of the articulated objects is known in advance. In our research, we applied a set of CNN-based methods that gradually estimate the state of the articulated object. The proposed approach does not require additional assumptions and extracts knowledge directly from training data.

2 Articulated Objects Detection and Estimation

The block diagram of the proposed method is presented in Fig. 2. The first neural network uses two pairs of RGB-D images to perform a segmentation of the rotation axis on an image. The result and the input RGB-D images are passed to the next model. This network estimates the depth value of the previously segmented axis. The third neural network estimates the configuration change of the articulated object (angle of rotation).

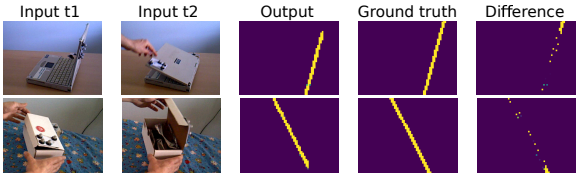


Fig. 3. Example output from the segmentation network compared to ground truth data. *Input t1* is registered at the beginning of the interaction and *Input t2* is the considered frame.

Fig. 4. Example scene visualizations. The red and blue lines represent the ground truth and estimated rotation axes, respectively.

2.1 Axis Segmentation

The CNN that performs the segmentation on the image uses the U-Net with ResNet34 as an encoder trained with the Dice loss. On the input of the CNN, we provide a differential image for the two RGB and depth images and the depth image registered at the beginning of the motion.
Example output of this network is presented in Fig. 3.

2.2 Estimation of the 3D Rotation Axis

The second network estimates a depth image that represents the points on the estimated axis of rotation. This step is needed because the depth from the depth images is not the same as the depth of the rotation axis. The CNN is based on the 3D U-Net [1] with a ResNet3D [6] as a encoder. We propose a new loss function that is a modified MAE loss but is only calculated on the segmented area. The input data is two RGB images, two depth images, and two outputs from the previous network. The output from the network is processed using the RANSAC algorithm, which allows us to extract the best two points that represent the axis in 3D space. Example outputs of the neural network and the obtained axis are presented in Fig. 4.

2.3 Rotation Angle Prediction

The last neural network predicts the joint state change of an articulated object. The used model is the ResNet34 which takes on the input a subtraction of the RGB images, subtraction of depth images, the depth image at the beginning of the motion, and the output from the previous network. We start the training using *Mean Squared Error* as a loss function and then switch to *Mean Absolute Error*.

Table 1. Average error of predicted angle values [rad] for test sequences from the RBO dataset

book10	book22	cardboardbox16	cardboardbox20	microwave12	microwave20	laptop13	laptop17
0.283	0.113	0.096	0.034	0.020	0.028	0.185	0.325

3 Tests and Results

To verify our method, we performed a series of experiments on the test sequences from the RBO Dataset. Taking the first frame of each sequence as a reference, we then iterate through the other frames in the sequence. The results of the tests are presented in Table 1.

4 Conclusion

In this paper, we propose a system that estimates the direction of a rotation axis and angle of rotation from a pair of RGB-D images using a set of neural networks. In contrast to classical methods [5] we do not assume that the surfaces of articulated objects are flat or the robot has a full 3D model of the scene. The average error for the test sequences is 0.135 *rad* (7.8°) which makes the method applicable on real robots. In the future, we are going to integrate the method into our mobile-manipulating robot and verify the system in real-life scenarios.

References

1. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. CoRR **abs/1606.06650** (2016)
2. Desingh, K., Lu, S., Opipari, A., Jenkins, O.C.: Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In: International Conference on Robotics and Automation. Montreal, Canada (2019)
3. Hofer, S., Lang, T., Brock, O.: Extracting kinematic background knowledge from interactions using task-sensitive relational learning. In: International Conference on Robotics and Automation (2014)
4. Martín-Martín, R., Eppner, C., Brock, O.: The rbo dataset of articulated objects and interactions. The International Journal of Robotics Research **38**(9), 1013–1019 (2019)
5. Staszak, R., Molska, M., Młodzikowski, K., Ataman, J., Belter, D.: Kinematic structures estimation on the RGB-D images. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1, pp. 675–681 (2020)
6. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. CoRR **abs/1711.11248** (2017), http://arxiv.org/abs/1711.11248
7. Wu, R., Zhao, Y., Mo, K., Guo, Z., Wang, Y., Wu, T., Fan, Q., Chen, X., Guibas, L., Dong, H.: Vat-mart: Learning visual action trajectory proposals for manipulating 3D articulated objects (2021)

Eldercare Robots in the Age of AI: Are We Ready to Address the User Needs?

Piotr Skrzypczyński[1], Sławomir Tobis[2]

[1]Institute of Robotics and Machine Intelligence,
Poznań University of Technology, ul. Piotrowo 3A, 60-965, Poznań, Poland
[2]Department of Occupational Therapy, Poznan University of Medical Sciences,
ul. Święcickiego 6, 60-781, Poznań, Poland
piotr.skrzypczynski@put.poznan.pl; stobis@ump.edu.pl

Abstract. This position paper presents an attempt to analyse the requirements of older users they formulate with respect to healthcare assistive robots to be used at home. This study is based on matching the hands-on experience with the use of a mobile robot at patients' homes within the ENRICHME project to the briefly sketched state-of-the-art in AI-based robotics. We consider three aspects of the robotics and AI technologies that are particularly important to the autonomy, safety, and user acceptance of eldercare robots: human-aware navigation, recognition of the user's state and actions, and natural means of man-machine communication. The outcome of our study identifies the key AI solutions that need to be integrated in such a robot, and challenges that still need to be addressed by research. This brief paper can serve as a background for discussion on how to take user-specific requirements into account in research and implementation of eldercare robots.

Keywords: Healthcare robot · eldercare · artificial intelligence

1 Introduction

The ageing society in Poland and the increasing number of people with prolonged diseases and disabilities (also as a result of the SARS-CoV-2 pandemics) poses an increasing problem. As the shortage of medical and care workers becomes evident, we need to rely more on assistive technologies [16]. However, these technologies still cannot address specific needs of older people. Not only if the robot has to physically interact with objects, but also whenever a more proactive behavior or conversational engagement of the robot is expected, for example during a dialogue with an older person with memory problems [18]. We thus formulate the thesis that the limited acceptance of existing assistive robots for older people at home is largely due to the mismatch between the communication interface capabilities of the robot, the needs of older adults and the limited ability of the robot to recognise situations requiring its own actions.

While robots have been developed that work as domestic helpers addressing the medical, physical or cognitive issues [4, 14, 16], and are even commercially

available, they are complicated and expensive. However, the last decade witnessed enormous progress in AI-based robotics that opened new possibilities, such as affordable 3-D sensing and powerful machine learning methods. Hence, in this short paper we provide a simple case study of the requirements by an average older person interested in having a robotic helper at home, and then we identify the AI and robotic technologies that are crucial to better acceptance of such a robot at home.

2 Do We Really Know the User Needs?

Important user experience concerning the use of an autonomous robot to support older people with mild cognitive impairment (MCI) was gathered in the ENRICHME project [13]. The project investigated user acceptance, attitude towards the robot, perception of the robot and the perceived levels of its naturalness, animacy and intelligence. Whereas the results showed high acceptance of the robot in the context of enjoyment and positive social influence, the perceived level of usefulness, and the social presence were scored somewhat lower, with the participants, who were more computer savvy being less afraid of using the robot.

In order to illustrate the exemplary needs of an older person, we present one case, already in the context of the new pandemic situation. Mrs. Zofia is 78 years old and lives alone in a large apartment block. Her children and grandchildren are in constant telephone contact with her. She has no friends in the area where she lives, so she has to cope with everything on her own. Before the pandemic she used to meet frequently with her friends, but now they only call each other for fear of COVID-19. Zofia does not use a computer. Although she has a modern smartphone, she does not know its functions. Zofia has suffered from hypertension, diabetes and hypothyroidism for many years. She takes a lot of medications, and sometimes forgets to take them at the right time, and then it happens that she takes the whole day's medication at once. She often looks for various items - such as her glasses or keys. Since the beginning of the pandemic she has felt lonely as she leaves the house less often. She is independent in her daily activities, she still does a little shopping herself and cooks for herself (however, not every day). Sometimes she forgets to eat the dinner she cooked the day before. She knows that if someone was with her, she would be more mobilised.

The user needs that are identified in this case study are as follows: (i) reminders about medication; (ii) reminders about food and drink; (iii) help with cooking – showing recipes and how to prepare them; (iv) physical exercise; (v) cognitive exercises; (vi) leisure activities, "to have someone over"; (vii) help with finding objects. One important observation is that these needs are mostly related to gathering and processing information, rather than physical help given by someone. Hence, they can be addressed by a robot that is mechanically simple (a manipulator arm is not needed), and therefore affordable. However, this robot must have advanced perceptual, cognitive and learning capabilities, in order to interact with the user and to recognise the semantics of the environment, as well as the meaning of the actions it needs to take.

3 Key AI and Robotics Technologies

Considering the presented case study, and the results of previous investigations concerning healthcare robots for the elderly in home environments [10, 14, 13], we select three areas of technological improvements that are crucial for addressing the needs of older users that have to live with autonomous robots as their companions and domestic helpers.

3.1 Human-aware Navigation in Home Environment.

A fundamental ability of an autonomous robotic companion is to navigate safely and accurately in home environment, which can be cluttered, semi-structured, and non-stationary. Accurate localisation that exploits laser, visual or RGB-D sensing is necessary to interact with objects at known locations. Recent algorithms combine learned object detection and/or semantic segmentation with Simultaneous Localization and Mapping (SLAM) techniques in order to provide high localisation accuracy in dynamic environments [9]. Global localisation [21] is necessary whenever the robot gets lost due to perception failures, occlusions, etc., or it starts operating in a new environment. A robot should plan its motion taking into account the information about the user activity to avoid collisions with humans. Also spaces, where activities are performed, and affordance spaces that denote potential activity spaces (e.g. an area next to a couch or chair) should be considered to determine what areas should not be traversed at the given moment [12]. The planning algorithms should also consider social conventions and the interpersonal zones that correspond to certain actions [18].

3.2 Recognition of the User's State and Actions.

A healthcare robot should observe the older user in order to respond quickly in case a dependent person goes through a distress situation. To implement this, the robot needs to identify humans, distinguish the user from others, and identify the pose of the person. Person detection is accomplished using 2-D laser, RGB-D, visual or thermal data, with best results provided by multi-sensor systems, e.g. the one used in ENRICHME [3] that combined laser, RGB-D and thermal perception. Recent person detection, pose estimation [19], and re-identification solutions are mostly based on deep neural networks [17] that outperform classic approaches by a large margin when implementing classification or semantic segmentation tasks [15]. One important issue for eldercare robots is fall detection, as for the older population, over one-third of falls lead to major injuries. While there are some fall detection techniques based on wearable sensors, vision-based methods are most appropriate for an autonomous robot [7]. These methods can recognise human pose, human movement or both, and detect a fall in case the established criteria are met. Convolutional or recurrent neural networks are used to detect falls, with the LSTMs being able to handle the dynamics in image sequences [7]. Detection and classification of potential obstacles can be also used to prevent falls by alerting the user before a dangerous situation occurs [6].

3.3 Information Processing for Efficient and Intuitive Communication.

The conversational abilities of a healthcare robot for older adults are central in the light of the identified user requirements. Conversation methods leverage Natural Language Processing (NLP) techniques [8], and a broad family of planning and reasoning algorithms [1]. Particularly relevant is the capability of producing contextualised interactions that may range from reminding healthy dietary suggestions [11], through helping to follow a rehabilitation programme at home, to supportive messages when it comes to react, for example, to an older person who is experiencing disorientation or memory problems [1]. The robot should respond actively to situations that were not pre-programmed by the developer; which is possible employing the Dual Process Theory [15], with an intuitive planning system that "thinks fast" whenever the response time is crucial and a deliberative component ("thinking slow") that optimises the plans. Machine learning should be leveraged to distill computation-intensive algorithms into learned policies to obtain a set of skill-specific experts that form the system of intuitive responses. It is important to engage the robot and user within a natural conversation with the content managed in real time [1]. Conversation based on speech seems to be a better way to attract the attention of an older user than employing an on-screen interface [18]. As the robot can gather user-specific data, including medical information and observational data, privacy and safety standards in processing of sensitive data must be carefully formulated and enforced [5].

Machine learning can also help in providing the older person with information and entertainment content gathered from the Internet, e.g. by implementing a customised recommendation mechanism and removal of fake news. The postulated ability to assist while preparing meals also can be implemented using machine learning techniques, as shown by the solution from [2], which suggests modifications to culinary recipes to address specific dietary requirements.

A much more challenging task is to find physical objects in the environment, to help users with memory or sight problems. This task was implemented in ENRICHME using RFID tagging [3], but can also be accomplished using visual sensing with learned object detectors and active perception [20].

4 Conclusion

This short position paper builds upon the previous research, mostly related to the ENRICHME project and published in the healthcare context, but confronts the identified requirements of older users with the recent AI and robotics developments, mostly based on the fast pace progress in machine learning. From this analysis we conclude that the key technologies in the areas of machine perception, state estimation (with respect to both the robot and the user) and planning are ready to be implemented in home healthcare robots. The remaining challenges concern mainly the more natural ways of communication based on speech processing and NLP, with the ability to learn the user's specific behavior and to use the information acquired from the external sources (e.g. the Internet) to help

the user in everyday activities, encouraging and advising the older person more like human companions do.

References

1. De Benedictis, R., Umbrico, A., Fracasso, F., et al A Two-Layered Approach to Adaptive Dialogues for Robotic Assistance. IEEE Int. Conf. on Robot and Human Interactive Communication (RO-MAN), (2020), 82–89.
2. Bień, M., Gilski, M., Maciejewska, M., et al.: RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation. Int. Conf. on Natural Language Generation, (2020), 22–28
3. Coşar, S., Fernandez-Carmona, M., Agrigoroaie, R. ENRICHME: Perception and Interaction of an Assistive Robot for the Elderly at Home. International Journal of Social Robotics, 12, (2020), 779-–805.
4. E. J. G. van der Drift, E. J. G., Beun, R., Looije, R., Blanson Henkemans, O. A., Neerincx, M. A.: A Remote Social Robot to Motivate and Support Diabetic Children in Keeping a Diary, ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI), (2014), 463–470.
5. Fosch Villaronga, E., Felzmann, H., Pierce, R., *et al.*: Nothing Comes between My Robot and Me: Privacy and Human-Robot Interaction in Robotised Healthcare. Data Protection and Privacy: The Internet of Bodies, Hart Pub. (2018).
6. Gruszczyński, D., Stefańczyk, M.: Active fall prevention: robotic vision in AAL. arXiv:2103.09298v2, (2021).
7. Gutiérrez, J.; Rodríguez, V.; Martin, S.: Comprehensive Review of Vision-Based Fall Detection Systems. Sensors, 21, 947, (2021).
8. Jay, C., Hareem, S., Katelyn, B., *et al.*: Case Report: Utilizing AI and NLP to Assist with Healthcare and Rehabilitation During the COVID-19 Pandemic. Frontiers in Artificial Intelligence, 4, (2021).
9. Ji, T., Wang, C., Xie, L.: Towards Real-time Semantic RGB-D SLAM in Dynamic Environments. IEEE Int. Conf. on Robotics and Automation (ICRA), (2021), 11175–11181.
10. Łukasik, S., Tobis, S., Kropinska, S., Suwalska, A.: Role of Assistive Robots in the Care of Older People: Survey Study Among Medical and Nursing Students. Journal of Medical Internet Research, 22, (2020).
11. Łukasik, S., Tobis, S., Wieczorowska-Tobis, K., Suwalska, A.: Could Robots Help Older People with Age-Related Nutritional Problems? Opinions of Potential Users. International Journal of Environmental Research and Public Health, 15, (2018).
12. Möller, R., Furnari, A., Battiato, B., Härmä, A., Farinella, G.: A survey on human-aware robot navigation. Robotics and Autonomous Systems, 145, 103837, (2021).
13. Piasek, J., Wieczorowska-Tobis, K.: Acceptance and Long-Term Use of a Social Robot by Elderly Users in a Domestic Environment. Int. Conf. on Human System Interaction (HSI), (2018), 478–482.
14. Robinson, H., MacDonald, B. Broadbent, E.: The Role of Healthcare Robots for Older People at Home: A Review. Int. J. of Social Robotics, 6, (2014), 575—591.
15. Roy, N., Posner, I. Barfoot, T, *et al.*: From Machine Learning to Robotics: Challenges and Opportunities for Embodied Intelligence. arXiv, 2110.15245, (2021).
16. Sapci, A. H., Sapci, H. A.: Innovative Assisted Living Tools, Remote Monitoring Technologies, Artificial Intelligence-Driven Solutions, and Robotic Systems for Aging Societies: Systematic Review. JMIR Aging, 2(2), (2019).
17. Tian, Z., Chen, S., Wang, D., Lu, J.: A Survey of Person Re-identification Based on Deep Learning. Int. Conf. on Computing and Pattern Recognition, (2021), 36—42.

18. Tanioka T, Yokotani T, Tanioka R, *et al.*: Development Issues of Healthcare Robots: Compassionate Communication for Older Adults with Dementia. International Journal of Environmental Research and Public Health, 18(9), 4538, (2021).
19. Wang, J., Sun, K., Cheng, T. *et al.*: Deep High-Resolution Representation Learning for Visual Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, 43(10), (2021), 3349–3364.
20. Wang, Q., Fan, Z., Sheng, Wh. et al.: Finding misplaced items using a mobile robot in a smart home environment. Frontiers Inf Technol Electronic Eng., 20, (2019), 1036—1048.
21. Wietrzykowski, J., Skrzypczyński, P.: PlaneLoc: Probabilistic global localization in 3-D using local planar features. Robotics and Autonomous Systems, 113, (2019), 160—173.

Onboard to Satellite Image Matching for Relocalization of the UAVs*

Tomasz Pogorzelski[1][0000−0002−8299−0727] and Karol Majek[2][0000−0002−1351−8496]

[1] Łukasiewicz Research Network – Institute of Aviation, 02-256 Warsaw, Poland
[2] Cufix, 05-825 Grodzisk Mazowiecki, Poland
karolmajek@cufix.pl

Abstract. In this paper, we study image-based localization as a component of a vision navigation system. We perform an experimental verification of image matching techniques on a real UAV flight in a rural area. Images from onboard camera are matched to the satellite images based on last known position. We find deep front-end SuperPoint with deep middle-end matcher SuperGlue better suited than the pretrained LoFTR for the image-based localization in feature-poor areas. The proposed component implementation allows UAVs for emergency localization based on single frame with error below 100m in a rural area.

Keywords: Image-based localization · UAV localization · Image matching.

1 Introduction and Related Work

We aim to ensure redundancy of satellite navigation in applications including acquisition of photographic materials with the use of UAVs [5]. The goal is to achieve a visual navigation component for the autopilot by matching the camera images to the orthophotomap for a large operation area. Image based localization for UAVs can be considered as full 6-dof [6] for indoor applications as well as for gps-denied outdoor environments [1]. Such approaches use conventional algorithms [7], either deep learning [7] or mixed [8]. For this purpose, the Mutual Information algorithm [12], image segmentation [3] and graph search are also often used [2]. For the purpose of image-based localization, hand-crafted image matching methods such as ORB [9] can be used. In contrast, newer approaches are based on neural networks and are learned from data during the training process. Use of neural networks is relevant to UAVs since availability of high performance processing on embedded devices such as Nvidia Jetson Xavier series or Intel Movidius. Feature detector and descriptor SuperPoint [4] was introduced as a deep front-end. SuperGlue introduced in [10] is a deep middle-end matcher which utilizes graph neural networks to solve assingment problem. Introduced recently, LoFTR is a detector-free method inspired by the SuperGlue matching [11] not limited to detected keypoints.

* Supported by an implementation doctorate registered at the Doctoral School of the Warsaw University of Technology and the Center for Unmanned Technologies, Łukasiewicz Research Network - Institute of Aviation.

2 Redundant Image Based Localization System for UAVs

To determine the absolute geographic location we use: the recently known position, the current image from the camera and the geo-referenced orthophotomap. The last known position is used to limit the the analyzed area - Region Of Interrest (ROI) in order to ensure the proper performance of the algorithm. Then the current camera image frame is compared with the cut ROI. The selected deep neural networks determine matches between images. Based on the best fit, a homography matrix, and a perspective quadrilateral are computed. The confidence of measurement depends on the number of detected points. The return is calculated from the middle of the top edge of the quadrilateral.

We compare learning-based feature detector and descriptor SuperPoint [4] with feature matcher SuperGlue [10], and detector-free LoFTR [11]. During system initialization, a rough start position is taken to limit the ROI (Region Of Interrest) search area from the whole map. Using the above algorithms, common points characteristic for the ROI and the current frame from the on-board camera are found. On the basis of these points, a homographic matrix is calculated, by means of which the perspective rectangle (blue) is calculated. Its geometric center is the location. Currently, the system analyzes each frame separately.

(a) SuperPoint + SuperGlue (b) LoFTR

Fig. 1: Visual comparison of image-based localization methods. Large rectangle is ROI used for localization. GPS trajectory in dark blue. Points in cyan - localization result. Cyan quadrilateral - represents field-of-view of the localized camera.

3 Experimental Results

The images generated by the system, based on a real UAV flight in a rural area, are shown below. For our input, conventional descriptor methods were only worked in simulated flight in Google Earth Studio, compared to a map with the same source. In order to get closer to the real conditions, a video was recorded made by a drone from a height of 120m above ground level. Flight logs were

used for the Ground Truth reference. In the case of a real flight recording, the above descriptors found too few common points. Only the use of deep learning techniques allowed for correct matching in this case. This may be related to unfavorable lighting, or to significant differences between the current image and the historical one from Google maps. The season was similar, so the colors of the vegetation did not matter. Figure 1 shows the visualization of the perspective (camera point of view) in the form of a blue quadrilateral placed on the map. Its geometric center is a preliminary location estimation. It can be noticed that both algorithms work worse in the case of obtaining a homogeneous image.

It has been observed that the even distribution of characteristic points in the image has a significant influence on the accuracy of the location. The method based on the average distance between the nearest, neighboring points was used to measure the uniformity coefficient. This tells whether the data points are concentrated or scattered. The graph below shows the error and the average distance between points in consecutive video frames. On Figure 2 a correlation between the calculated location error and the average distance of the point from the nearest neighbor can be observed.

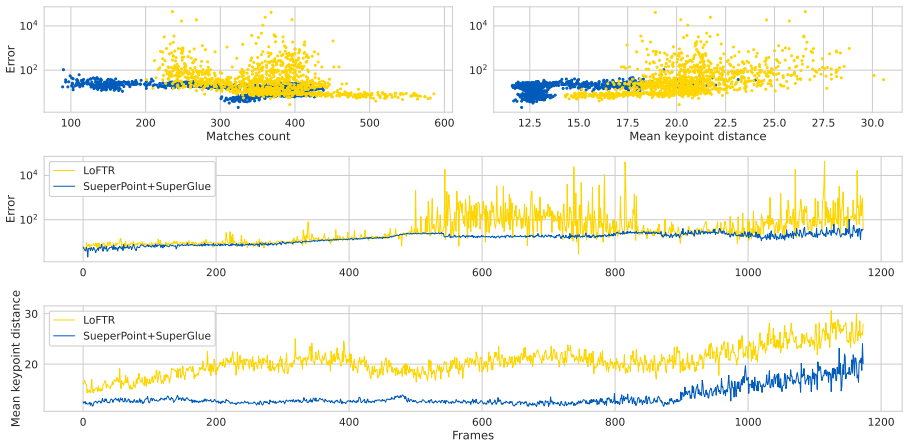


Fig. 2: 2D Error in meters related to the number of good matches and mean keypoint distance for SuperPoint with SuperGlue and LoFTR.

4 Conclusion and Future Work

We compared deep learning-based image matching methods for the purpose of UAV localization. Results show smaller error for keypoint-based method SuperGlue in favor of keypoint-free LoFTR pretrained in this setup. All correct matches were used to calculate the homography matrix. In the future, it is

planned to select only those with the greatest probability of correctness. Fine-tuning LoFTR on aerial images should improve the results. The proposed image-based localization module works on a single frame against a large ROI. Therefore, localization error on raw outputs is too large for UAV navigation purposes and should be fused with IMU and visual odometry to achieve a good accuracy in gps-denied environments. The described work concerns the analysis of a short video recording on a small area, however, it allows for further orientation of the work in an optimal way.

References

1. Ashraf, S., Aggarwal, P., Damacharla, P., Wang, H., Javaid, A.Y., Devabhaktuni, V.: A low-cost solution for unmanned aerial vehicle navigation in a global positioning system–denied environment. International Journal of Distributed Sensor Networks **14**(6), 1550147718781750 (2018)
2. Chiu, H.P., Das, A., Miller, P., Samarasekera, S., Kumar, R.: Precise vision-aided aerial navigation. IEEE International Conference on Intelligent Robots and Systems pp. 688–695 (09 2014). https://doi.org/10.1109/IROS.2014.6942633
3. Choi, J., Myung, H.: Brm localization: Uav localization in gnss-denied environmentsbased on matching of numerical map and uav images (08 2020)
4. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
5. Kacprzak, M.: Zastosowanie odbiorników gps i inercyjnych jednostek pomiarowych w akwizycji materiału zdjeciowego z wykorzystaniem bezzałogowych statków latajacych (bsl) pp. 267–276 (2016)
6. Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M.: Real-time image-based 6-dof localization in large-scale environments. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 1043–1050. IEEE (2012)
7. Mughal, M.H., Khokhar, M.J., Shahzad, M.: Assisting uav localization via deep contextual image matching. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 2445–2457 (2021). https://doi.org/10.1109/JSTARS.2021.3054832
8. Nassar, A., Amer, K., ElHakim, R., ElHelw, M.: A deep cnn-based framework for enhanced aerial imagery registration with applications to uav geolocalization pp. 1594–159410 (2018). https://doi.org/10.1109/CVPRW.2018.00201
9. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011)
10. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
11. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
12. Yol, A., Delabarre, B., Dame, A., Dartois, J.E., Marchand, E.: Vision-based absolute localization for unmanned aerial vehicles pp. 3429–3434 (2014). https://doi.org/10.1109/IROS.2014.6943040

Remote Control Architecture for Autonomous Electric Minibuses – Preliminary Study

Mateusz Kosior[0000-0003-4368-5364], Piotr Przystałka[0000-0002-3412-4533] , Wawrzyniec Panfil[0000-0001-7304-554X] , Wojciech Moczulski[0000-0002-4697-1561] , Maciej Malczyk[0000-0001-9299-1937] , Witold Krafczyk[0000-0003-4193-796X]

Silesian University of Technology, Gliwice, Poland
mateusz.kosior@polsl.pl

Abstract. The paper deals with a remote control architecture dedicated for autonomous electric minibuses. The proposed architecture can be used to control the vehicle in the direct mode or with the use of a semi-autonomous functionality. The main objective of the paper is to study the performance of LTE network in the task of remote control of the vehicle. The preliminary tests were carried out in a relevant environment. The obtained results proved that LTE network can be successfully applied as a standard of wireless broadband communication for remote control of vehicles with a semi-autonomous mode.

Keywords: electric cars, autonomous minibuses, remote control, wireless communication.

1 Introduction

Autonomous buses have become increasingly popular in recent years [1]. These can be seen as the mean to revolutionize transportation and to change the whole paradigm in the near future. The autonomy is divided into five levels by the Society of Automotive Engineers SAE J3016 [2], where Level 0 deals with no autonomy at all, whereas Level 5 corresponds to full autonomy in any conditions. This is no longer a problem to show examples of low-speed autonomous buses operating up to Level 4, see e.g. [3]. However, self-driving in Level 4 is supported only to limited operational design domain [2]. In this mode, the vehicle must be able to safely abort the trip, e.g. slow down and park the car, if the driver does not retake control. When the vehicle cannot return to the self-driving mode, the further acting of the vehicle is not possible without helping of the driver. This is a more difficult issue in a case when there is no driver inside the car. Such scenarios usually need a teleoperation mode which enables the vehicle to be taken over in case human supervision is needed. This paper is focused on communication issues related to the remote control architecture dedicated for autonomous electric minibuses.

2 Remote Control Architecture

The proposed configuration of the hardware components is shown in Fig. 1. The remote control system consists of a remote operator in charge of driving the vehicle; a PC-class computer that acts as a server with a fixed public IP and has indirect access to the LTE network via WAN; steering wheel and control pedals for generating control signals by the remote operator (not shown in the picture); monitors presenting the image from the camera(s) on the vehicle; TELTONIKA RUTX12 router with variable, public and shared IP, access to LTE network and a set of antennas (not shown in the picture); NRU-120S on-board computer for communication via the ROS network and video streaming; PCAN-USB Pro FD analyzer enabling preview and interaction with the CAN 2.0 bus (not shown in the picture).

The components of the remote control system are connected to an internal VPN network. The on-board computer is also connected to the MicroDAQ device, which is the gateway in communication with the other modules on the vehicle (not shown in Fig. 1). The system is currently under development, therefore the research presented in this article was carried out using an NVIDIA Jetson AGX Xavier Development Kit and an LTE mobile phone access point on the vehicle side, instead of the NRU-120S on-board computer and TELTONIKA RUTX12 router.

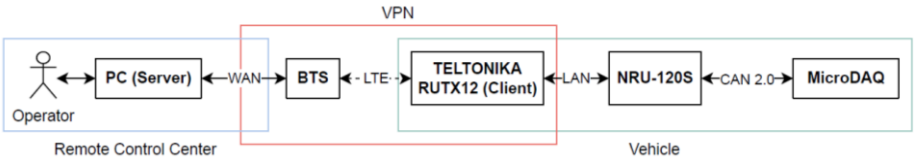


Fig. 1. Proposed Remote Control Architecture

3 Preliminary Results and Discussion

The communication system of the remote-control system was verified in a relevant environment (TRL6). The tests compared data transmission latency and frame loss measured for two LTE providers (a and b) with reference measurements made with Ethernet and wi-fi interfaces. A stationary PC connected to the internet and located at the Faculty of Mechanical Engineering of Silesian University of Technology (Gliwice, Poland) was the sender.

During the verification study two types of data were sent via TCP: (1) a stable video stream from a web camera and (2) ROS-based control frames. Latency was measured by comparing the timestamps embedded in the data frames with current time on the receiver. Image acquisition and display latencies, as well as any possible control input latencies were not considered. The receiver was placed in Świętochłowice (SW), in Czechowice (CZ) and then in Wielowieś (WW) in Poland. Receiver and sender were synchronized to a custom NTP server running on the sender PC.

Video was streamed from a web camera connected to the PC. The stream was encoded using OpenCV and JPEG with 90% quality. Image resolution was 1280x720 pixels at 30 FPS. Table 1 summarizes chosen results. Q1-3 represent the first, the second and the third quartile. Video was streamed outside of ROS network.

Table 1. Video streaming results for different interfaces

#	Interface	Receiver Location	Latency [ms] Min	Max	Avg	StDev	Q1	Q2	Q3	Frames lost
1	Ethernet	SW	17.7	298.3	22.9	28.3	18.2	19.1	19.6	0.0%
2	Wi-fi	SW	24.6	1767.2	166.2	31.2	34.2	89.9	362.0	0.0%
3	LTE (a)	SW	28.2	698.1	70.0	101.2	36.7	41.5	50.6	0.0%
4	LTE (b)	SW	22.1	1761.5	352.0	412.4	36.7	121.2	652.2	0.0%
5	LTE (a)	CZ	29.8	449.3	44.9	44.1	33.9	38.5	41.9	0.0%
6	LTE (b)	CZ	35.9	756.2	61.9	100.2	38.9	43.1	46.1	0.0%
7	LTE (a)	WW	18.7	431.9	34.4	21.7	25.0	29.5	49.4	0.0%
8	LTE (b)	WW	1.5	1312.7	97.6	277.8	3.6	5.8	12.1	0.0%

Control signals were simulated using timestamped ROS messages of types Empty, Bool, Int, UInt, Float and String. They were sent concurrently on different topics at 1 Hz rate. The measured latency was the difference between receive time and the timestamp embedded in the message. Chosen results are presented in Table 2.

Table 2. Control frames received using different interfaces

#	Interface	Receiver Location	Latency [ms] Min	Max	Avg	Frames lost
1	Ethernet	SW	22.3	18.3	20.5	0.0%
2	Wi-fi	SW	18.4	832.4	155.0	0.0%
3	LTE (a)	SW	20.2	150.0	66.6	0.0%
4	LTE (b)	SW	15.7	256.9	142.9	0.0%
5	LTE (a)	CZ	0.5	279.6	144.9	0.0%
6	LTE (b)	CZ	35.9	756.2	61.9	0.0%
7	LTE (a)	WW	9.0	218.3	144.8	0.0%
8	LTE (b)	WW	12.1	140.8	81.8	0.0%

Additionally, another experiment was conducted by measuring video latency while driving from Czechowice to Gliwice with the receiver onboard. The vehicle was moving approximately at speed of 30 – 40 km/s. Only LTE (provider b) was used.

Less dense areas far from city centers featured significant reduction in latency due to clear path to BTSs and less noise, while Świętochłowice provided the largest latency. The performance measured for both LTE providers differs much between the

sites. Dynamic test also highlighted bottlenecks in areas with no BTS coverage, such as forests. Nevertheless, about 70% of the video frames were received within 70 ms.

4 Conclusion

This paper shows some partial results of the study concerned with communication performance of the proposed remote control architecture. The main objective was to explore the applicability of LTE network for remote control of the vehicle. The approach was verified in a relevant environment (TRL6). It was initially proved that LTE network can be successfully applied as a standard of wireless broadband communication for remote control of vehicles equipped with a semi-autonomous driving mode. Moreover, some limitations have to be taken into account (i.e. areas with no BTS coverage, such as forests, or even zones where a change between BTSs is required) to create the final solution dedicated for real world conditions (TRL9). The more detailed analysis is needed taking into account existing solutions i.e. provided by Freedom Robotics [4] or DriveU [5] in order to identify pros and limitations of the proposed architecture.

Acknowledgements. The results presented in this paper were obtained in the frame of the project titled “Autonomous electric microbus adapted to move in platooning mode with intelligent passenger information system” - the grant agreement No. UDA-RPSL.01.02.00-24-0498/19-00 financed by RPO WSL. The project is carried out in the consortium of Blees LLC and Silesian University of Technology. This publication is partially supported from the statutory funds of Department of Fundamentals of Machinery Design.

References

1. Badue, C., Guidolini, R., Carneiro R.V., Azevedo, P., Cardoso, V.B., Forechi, A., Jesus, L., Berriel, R., Paixão, T.M., Mutz, F., De Paula Veronese, L., Oliveira-Santos, T., De Souza, A.F.: Self-driving cars: A survey. Expert Systems with Applications 165 (2021). doi: 10.1016/j.eswa.2020.113816.
2. On-Road Automated Driving (ORAD) committee: J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, Tech. Rep., (2021). doi: 10.4271/j3016_201609.
3. Kalda, K., Sell, R., Soe, R.-M.: Use case of Autonomous Vehicle shuttle and passenger acceptance analysis. Proceedings of the Estonian Academy of Sciences 70(4), 429–435 (2021). doi: 10.3176/proc.2021.4.09.
4. Freedom Robotics Homepage, https://www.freedomrobotics.com, last accessed 2022/03/29.
5. DriveU Homepage, https://www.driveu.in, last accessed 2022/03/29.

Robot Programming Interface with a Neural Scene Understanding System *

Bartłomiej Kulecki[0000−0002−2820−8212] and
Dominik Belter[0000−0003−3002−9747]

Institute of Robotics and Machine Intelligence, Poznan University of Technology,
ul. Piotrowo 3A, 60-965 Poznan, Poland
bartlomiej.kulecki@put.poznan.pl

Abstract. In this paper, we deal with the problem of robots programming using natural voice and gestures. We utilize voice recognition, objects detection, and human pose estimation modules to implement a new human-robot interface that allows defining the motion of the robot. The proposed interface enables robot programming without specially designed hardware interfaces. Finally, we present the efficiency of the proposed interface compared to the standard programming method.

Keywords: Human-robot interface · gesture recognition · 3D perception

1 Introduction

Cooperative robots become more popular in the modern industry where they support the production process. Once programmed, they perform the repetitive task for days or months. Re-programming the robot is a process that requires specialized domain knowledge. New robot programming interfaces enable operators to program the robot to perform complex tasks. In this research, we explore the problem of programming the cooperative robot using natural language. The main goal of the system is the interpretation of the operator's intentions from the camera images and voice commands.

Convolutional Neural Networks advanced scene understanding. To make robots more flexible and enable fast and natural motion programming, we propose the application of 3D perception and CNN for context understanding. We mimic interpersonal communication that is based on words and gestures. The operator uses his hands to select the object for grasping or asks the robot to pass the object to his hand. The voice commands trigger defined actions of the robot.

Most human-robot interaction systems take advantage of gesture recognition [6]. Neural networks are commonly used for hand/human pose detection and gesture classification [6]. The recognized gestures are later mapped with commands that correspond to the defined motion of the robot. Also, voice commands

* This work was supported by the National Centre for Research and Development (NCBR) through project LIDER/33/0176/L-8/16/NCBR/2017 and Poznan University of Technology project number 04/45/SBAD.

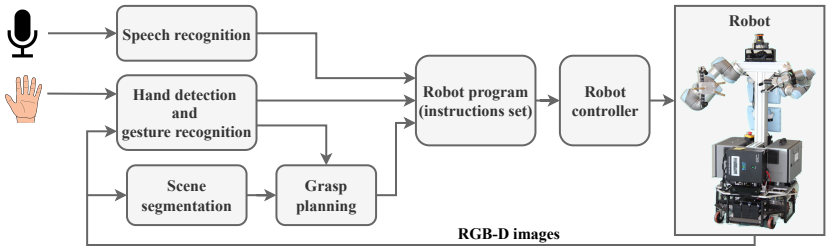


Fig. 1: Architecture of the proposed robot programming system

simplify the communication between the human operator and the robot [1] but the visual scene understanding enables the robot to interact with *a priori* unknown and continuously changing environment [1, 7]. The recognized commands can be later used to directly perform or schedule the repetitive tasks [2, 5]. In this paper, we integrate hand gestures recognition, voice commands with scene understanding to implement a new natural human-robot programming interface.

2 Natural Human-Robot Interaction

The architecture of the proposed system is presented in Fig. 1. The user interacts with the robot by gestures and voice commands. RGB-D images are utilized to determine the hand configuration and perform scene segmentation. Recognized voice commands, hand poses, and objects are used to plan the robot's motion.

2.1 Hand Detection and Gesture Recognition

The hand detection on the RGB image is performed using the state-of-the-art system for hand detection and tracking - Mediapipe Hands [8]. The output from the module is a set of 21 hand landmark points, consisting of 3 coordinates: x, y, and relative depth. We propose a method that recognizes gestures to allow effective interaction between the user and the robot. We defined two gestures represented by class names: "pointing" and "outreached". The set of 63 coordinates obtained from the hand detector is used as a feature vector for classification. First, we collected the training data set consisting of 9700 samples (5500 for the first and 4200 for the second class). Next, we trained four classification models: Random Forest, Support Vector Classifier, Ridge Classifier, and K-Nearest Neighbors. The final system utilizes the K-NN algorithm that has 99% accuracy on the test dataset containing 2400 samples. The recognized gesture type defines the robot's behavior. If the predicted class is "pointing", the program searches for the selected object. If the recognized gesture is "outreached" the target pose for object release is defined in the operator's hand.

2.2 Scene Segmentation and Grasp Planning

The 3D perception module performs scene segmentation based on point cloud processing [4]. The output of this module is a set of 3D-oriented bounding boxes

representing detected objects. The grasp planning algorithm [4] utilizes information about the current gesture and pose of the hand to find the object selected by the user. In this process, we use the position of two landmarks detected on the image [3], named INDEX_FINGER_MCP and INDEX_FINGER_TIP. Based on the depth image and camera matrix, we obtain the 3D position of these points and create a pointing vector P. In the next step, the root of the finger (MCP point) is connected with the objects' centers to obtain a set of M vectors. To decide which item is selected, we find the object (its id) with the minimum distance (smaller than the threshold value) to the pointing vector:

$$\arg\min_{id} \frac{\left\| \overrightarrow{P} \times \overrightarrow{M(id)} \right\|}{\left\| \overrightarrow{P} \right\|}. \tag{1}$$

2.3 Voice Commands

In the proposed system, the voice commands are used to confirm intentions expressed with a gesture. Our system [5] utilizes an of-the-shelves module - Google Speech Recognition to convert speech to text. Example commands used during experiments are:

1) ***grasp*** - the robot executes the trajectory to grasp the selected objects,
2) ***give to hand*** - the robot executes the trajectory to the position over the outreached hand and opens the gripper,
3) ***go home*** - moves the robot to the predefined configuration (gripper tilted down).

3 Results

The proposed system was tested in two scenarios (Tab. 1). During the first experiment, the robot was programmed using simple commands like *"open/close the gripper"* and *"go to initial configuration"*. If the *"go there"* command is recognized, the robot moves to the position of the detected hand. In the second experiment, the robot grasps the selected object. The example motion execution is presented in Fig. 2. During the experiments, we measured the programming and the execution time. We compared the proposed method with the programming using the visual marker, selecting objects by name (with objects detection), and traditional programming methods (teach pendant and teaching by doing) [5]. The results are presented in Tab. 1. The results show that programming the robot with gestures is more time-consuming than using the pointer but is more intuitive and does not require additional equipment.

4 Conclusions and Future Work

In this paper, we propose a system that utilizes voice commands, hand gestures, and 3D perception of the robot to program the execution of complex tasks. The advantage of the proposed system is the lack of additional hardware equipment like markers or pointers. Another benefit is that we can choose any type of object,

Table 1: Comparison of average time per one command in programming sequences for various programming methods.

test case	average time per one command [s]				
	teach pendant [5]	TbD [5]	pointer and voice [5]	object detection and voice [5]	gestures and voice
1	15.4	9.7	18.2	-	19.7
2	-	-	20.7	22.3	22.7

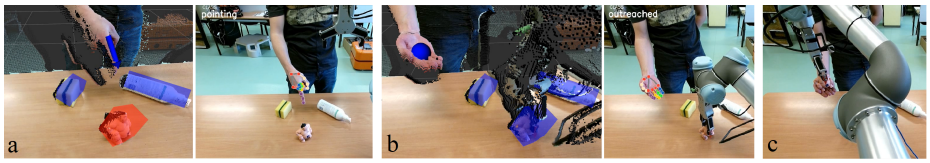


Fig. 2: Example point clouds and images from the camera during the grasping experiment: the user selects the object using the finger (a), the robot grasps the object (b) and passes it to the detected hand (c).

in contrast to the system with object detection, which allows us to select only objects from the training set. In the future, we are going to extend the number of recognized gestures to simplify robot programming.

References

1. Adamini, R., Antonini, N., Borboni, A., Medici, S., Nuzzi, C., Pagani, R., Pezzaioli, A., Tonola, C.: User-friendly human-robot interaction based on voice commands and visual systems. In: 2021 24th International Conference on Mechatronics Technology (ICMT). pp. 1–5 (2021)
2. Dudek, W., Winiarski, T.: Scheduling of a robot's tasks with the tasker framework. IEEE Access **8**, 161449–161471 (2020)
3. Google_LLC: Mediapipe - hand landmark model. https://google.github.io/mediapipe/solutions/hands.html, [accessed 4-03-2022]
4. Kulecki, B., Młodzikowski, K., Staszak, R., Belter, D.: Practical aspects of detection and grasping objects by a mobile manipulating robot. Industrial Robot **48**(5), 688–699 (Jan 2021)
5. Kulecki, B.: Intuitive robot programming and interaction using RGB-D perception and CNN-based objects detection. In: Szewczyk, R., Zieliński, C., Kaliczyńska, M. (eds.) Automation 2022: New Solutions and Technologies for Automation, Robotics and Measurement Techniques. Springer International Publishing (2022), (in print)
6. Mazhar, O., Ramdani, S., Navarro, B., Passama, R., Cherubini, A.: Towards real-time physical human-robot interaction using skeleton information and hand gestures. In: IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS). pp. 1–6 (2018)
7. Park, K.B., Choi, S.H., Lee, J.Y., Ghasemi, Y., Mohammed, M., Jeong, H.: Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality. IEEE Access **9**, 55448–55464 (2021)
8. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L., Grundmann, M.: Mediapipe hands: On-device real-time hand tracking (2020)

System for Detection and Tracking of Windows in Urban Environment *

Krzysztof Krygiel[1][0000−0003−0017−0299], Karol Majek[1][0000−0002−1351−8496], and Janusz Bedkowski[2][0000−0003−2630−1947]

[1] Mobile4.pro sp. z o.o., 02-621 Warsaw, Poland;
karol.majek@mobile4.pro
[2] Institute of Fundamental Technological Research, Polish Academy of Sciences, 02-106 Warsaw, Poland

Abstract. In this paper, we propose a system for tracking window edges, which can be used to improve computer vision tasks such as visual odometry in urban scenes. We use deep neural network based detector to detect windows as aligned bounding box predictions. In the second step, we track windows using keypoint-based tracking. We analyze the graph of detected matches across multiple frames to remove outliers. In the final step, we use learned line detector to refine the axis aligned bounding box approximation of the window. Each component of the system can be improved to achieve better results of the system.

Keywords: Window detection · tracking · image matching.

1 Introduction and Related Work

In urban environments such as large cities, repeating patterns such as windows occur frequently. In multi-store apartment buildings windows are tessellated which could lead to assignment errors in computer vision methods such as visual odometry or structure from motion. In this paper, we focus on apartment windows in particular and propose a system which can detect, track and refine detected windows to provide reliable information across multiple frames.

Window detection can be performed from aerial images using instance segmentation technique [5] as well as from ground imagery [9]. It is one of the tasks in building facades parsing task [7] which is currently solved with semantic segmentation and instance segmentation, deep learning based methods [9]. Windows are annotated in Open Images dataset [3] and for the purpose of experiments in this work we use a pretrained model on this dataset by [10]. Several methods can be used for tracking detected windows starting from tracking by detection using Intersection over Union (IoU) with boxes from subsequent frames requires

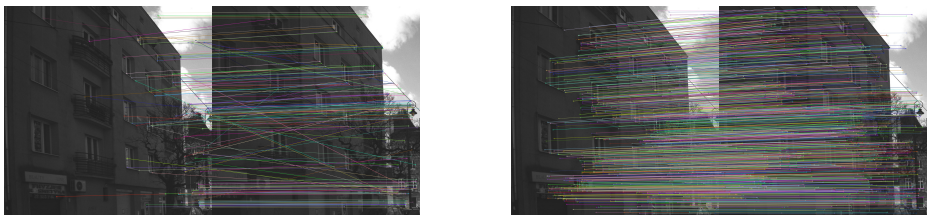
* The research leading to these results has received funding from POIR.01.01.01-00-0494/20 "Development and verification of the automatic location and 3D visualization of the selected objects in urban environment technology together with people flow modeling".

a guarantee of small movements of objects in the image [1]. Image matching state of the art was significantly improved by SuperPoint network [2] compared to hand-crafted detectors and descriptors. This feature matching deep front-end was later improved by introducing deep middle-end matcher SuperGlue [12]. It introduced Graph Neural Networks to solve the assignment optimization problem. Recently, a detector-free method was introduced to solve image matching - LoFTR [13] which uses Transformer architecture with global receptive field to provide dense matches at coarse level and refine only good ones.

In our work we do not aim to parse facades accurately, instead we want to provide good matches between images from cities where tessellated structures such as windows occur. Therefore, we propose a modular system for detection, and tracking of windows.

2 Proposed System

We propose a system which consists of 4 steps: axis-aligned bounding box detection, tracking using image matching, graph-based outlier removal, and finally refinement based on-line detection. In first step we perform detection using axis aligned window detection method based on YOLOv3 [11] trained on Open Images dataset [3]. In the second step we aim to match windows in consequtive frames. Our approach is to use keypoints-based image matching methods in order to solve association problem. Methods such as *ORB* or *SIFT* turned out to be not precise enough to track windows as shown in Figure 1a. We confirmed that the *SuperPoint* method for generating characteristic points and *SuperGlue* [12] for connecting them is sufficient in our case. Figure 1b presents *Superglue* result for the same pair of pictures.



(a) SIFT [8] (b) Superpoint [2] + Superglue [12]

Fig. 1: Keypoint matching results for object detection task; hand-crafted detector compared to learned method

The goal of window tracking is to assign an unique identifier to each physical window. Therefore, errors can be of two types: same identifier may by assigned to different windows, single window observed in many photos may receive more than one identifier. In order to detect which matches are invalid, it is worth of transforming that problem using graph theory. Each vertex is a single bounding

box detected by the neural network. Edges connect vertices only when bounding boxes were matched, considering an appropriate number of matching key points as a condition. If all matches were correct, then the graph would contain many connected components and each component would represent a single physical window.

Such component should contain a lot of edges from single vertex because single bounding box should match with corresponding boxes from neighboring frames. Wrong matches are converted to edges, which create a single component from two components which should remain separate. If there is just one such edge, it is called a *bridge* - edge of a graph whose deletion increases the graph's number of connected components. It is possible to find bridges in the graph to eliminate invalid matches. Due to the fact that errors might be a bit more frequent it is good to look for components such that even if 2 edges are removed it still remains a component. Each *3-edge-connected component* should contain all vertices representing bounding boxes from different images with same physical window inside. After 3 steps of the system, the result is presented in Figure 2. In the last step, we apply line detection method such as Deep Hough-Transform Line Priors [4] or HAWP [14] to refine the bounding boxes.

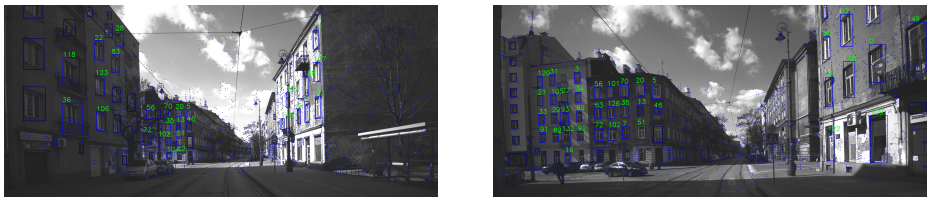


Fig. 2: Preserved window identifiers using keypoint-based tracking in two frames captured from distant positions with assigned identifiers. Axis aligned window detection method based on YOLOv3 [11] trained on Open Images dataset [3] from [10]

3 Conclusion and Future Work

We presented a system for detection and tracking window edges. Proposed system can track similar objects reducing the number of mismatched windows thanks to used graph representation. We identified two problems which are solved thanks to image matching: tracking obscured windows invisible, and not detected in several frames, and when there is an object which partially covers the window in one frame and another window in the second frame, there might exist enough matches between points on that obscuring object and those bounding boxes may be connected. Thanks to the modularity of the proposed system, we identify multiple ways of further improving the results of detection, assignment, and tracking of windows. Detection module can be replaced with

recent state-of-the-art method [6]. Window edges detection may be improved by preparing dataset containing images captured from the city and using them to train network detecting wireframes. The system output can be improved when use LoFTR image matching [13]. Tracking module can be improved with recent methods such as [15]. The interesting next step would be to use Graph Neural Networks such as in [12] to eliminate bad matches. Evaluation of the proposed system on a visual odometry task on publicly available datasets is planned a future work, as well as involving multi object tracking metrics.

References

1. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). pp. 1–6. IEEE (2017)
2. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
3. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4. International Journal of Computer Vision **128**(7), 1956–1981 (3 2020). https://doi.org/10.1007/s11263-020-01316-z, http://dx.doi.org/10.1007/s11263-020-01316-z
4. Lin, Y., Pintea, S.L., van Gemert, J.C.: Deep hough-transform line priors (2020)
5. Lippoldt, F., Erdt, M.: Window detection in aerial texture images of the berlin 3d citygml model. arXiv preprint arXiv:1812.08095 (2018)
6. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. arXiv preprint arXiv:2111.09883 (2021)
7. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. ACM Trans. Graph. **26**(3), 85 (2007)
8. Ng, P.C., Henikoff, S.: Sift: Predicting amino acid changes that affect protein function. Nucleic acids research **31**(13), 3812–3814 (2003)
9. Nordmark, N., Ayenew, M.: Window detection in facade imagery: A deep learning approach using mask r-cnn. arXiv preprint arXiv:2107.10006 (2021)
10. Osmulski, R.: Yolo open images (2018), *https* : *//github.com/radekosmulski/yolo*$_o$*pen*$_i$*mages*
11. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018)
12. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks (2020)
13. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
14. Xue, N., Wu, T., Bai, S., Wang, F.D., Xia, G.S., Zhang, L., Torr, P.H.S.: Holistically-attracted wireframe parsing (2020)
15. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021)

AI-Based Design of Decision Support Systems for Industrial Risk Management

Andrzej M.J. Skulimowski,[1,2] [0000-0003-0646-2858] Paweł Łydek[1,2] [0000-0001-9077-4998]
[1]AGH University of Science and Technology, Chair of Automatic Control and Robotics, Decision Science Laboratory, 30-059 Kraków, Poland
[2]International Centre for Decision Sciences and Forecasting, Progress & Business Foundation, 30-048 Kraków, Poland
ams@agh.edu.pl, lydek@agh.edu.pl

Abstract. This paper presents a new approach to design intelligent decision support system for industrial risk management. The need for this class of systems has been driven by complex resilience building and action planning problems that occur in large industrial plants. The proposed software architecture of the decision support system (DSS) is AI-based and applies Bayesian, causal and anticipatory networks, as well as multicriteria analysis, expert information fusion and knowledge processing techniques. The use of AI-tools follows the AI-alignment paradigm, where AI evolution is followed to discover most suitable techniques to solve safety-related problems. We propose a general scheme of DSS for risk management that includes threats, sensors, information processing, and decision models. The DSS and risk management module are coupled within a semi-supervised machine learning procedure, where the results of prior decisions serve to learn risk mitigation action parameters and managerial preferences.

Keywords: Intelligent Decision Support Systems, Risk Modelling, Multicriteria Optimization, Semi-Supervised Learning, Information Fusion.

1 Introduction

Risk management is a broad subject which often involves the use of artificial intelligence (AI) and machine learning (ML) methods. It usually focuses on financial risks or cyber threats. This article refers to a real-life case of an intelligent decision support system (DSS) designed for a limestone mine in Poland, where heterogeneous natural and anthropogenic threats can be prevented or mitigated with a wide spectrum of AI-based techniques. The heterogeneous character and high frequency of acquired threat data make it impossible to process all this information solely by human first responder forces in case of emergency. Moreover, while some of the sensor information, such as fire or flood alerts, can be quickly transformed into mitigating actions, other information may be incomprehensible fully or in part for the security staff of the enterprise.

AI methods are also the basis to design the managerial response with the DSS. To deal with complex information and knowledge management problems that should ultimately lead to optimal resolution of risk management issues and minimize the related losses, we propose and construct a causal model of threats, risks, crisis management decisions, and their consequences. The intelligent DSS that incorporates this

model is capable of recommending situation-dependent risk mitigation actions, operations, and strategies to ensure an optimal level of industrial safety. Such systems will be termed industrial risk management DSS or disaster resilience management DSS (IRM DSS or DReMSS [8]). The ultimate model should be implemented in an IRM DSS that can support decisions at all relevant levels, from immediate remedies to planning complex operations and long-term strategic resilience-building measures. The system is developed according to the DevOps paradigm, while experience learned with its operations is additionally enhanced by links to external AI-foresight and AI-alignment modules. These support the future-oriented development of the next IRM DSS releases. The AI techniques are applied primarily to sensor information processing and fusion, design of intelligent multicriteria DSS and ML procedures. Optimal decisions are derived with the core decision support engine that processes all threat-related information available, such as sensor data, historical facts on past threats, and the ways and results of their handling. The constraints on the decision rules are imposed by law or by internal regulations in the industrial plant concerned.

In the industrial security problems considered in this paper, risk is attributed to external threats, to information processing procedures that can bias the data with errors, to human operational errors, and to systematic erroneous decisions that can be made during risk management. For example, temperature measurement in fire sensors can be biased by the inaccuracy of thermometers, which can be assessed in advance as ex-ante measurement error. If the uncertainty is too high, the information from additional remote infrared sensors can be fused to yield trustworthy risk assessment.

The general threat transfer will be modelled as a network, where incidental operational and decision-making faults are sources of additional risks. This network is complemented by the risk management and optimization model involving decision algorithms, actions, and actuators. Both components of the model are coupled by feedback information received by the sensors, compared to the values provided by the model and presented to the ML module and to the decision makers. The DSS engine uses the above model components sequentially and links them to a semi-supervised machine learning procedure, where the assessments of previous decisions serve to learn sensor characteristics, risk mitigation procedure parameters, and managerial preferences.

2 Real-Life Needs And Related Research Inspiration

Limestone mining is related to specific risks caused by the occurrence of phenomena such as landslides or rock falls in multiple locations distributed over a large area. Such phenomena result from heavy machine traffic, shooting work, or from natural causes such as heavy rain or snowfall. Therefore, safety management systems implemented in an opencast mine should ensure a periodic inspection of the mining area in order to identify potential threats. Due to the nature of the business, i.e. working on a vast and rapidly expanding area, physical securing of operations with fences or human patrolling are practically impossible. The motivation for applying advanced AI information processing methods and software originates from the fact that the above challenge can be satisfactorily met by AI-based monitoring and prevention technologies. These include visual monitoring cameras, radars, lidars, and other sensors

installed on autonomous ground-based or aerial inspection robots. The ML-based automatic interpretation of images from the protected area will indicate sites at risk, e.g. those with displacement of geological structures, where rock is likely to collapse.

While AI is widely used in financial risk management systems, its deployment in natural and anthropogenic crisis management systems was rare until recently [8]. When designing the IRM DSS for the limestone mine that serves as an illustration of our approach, we took into account the current development of crisis management systems with decision support functionalities. This class of DSS emerged from early warning systems that evolved into cloud-based heterogeneous signal processing [1]. Then, various crisis management and DSS architectures were proposed [2]. The diversity of threats and feasible prevention and mitigation measures showed an increased relevance of trust analysis of information sources [6]. The evacuation of staff and equipment turned out to be the most important problem from the point of view of solving the resilience problems in the opencast mine related to natural threats [3]. The hierarchical character of emergency decision making and long-term prevention planning resulted in the adoption of the roadmapping methodology to design the DSS [4].

The main goal of this article is to propose a software architecture that allows decision makers responsible for crisis management to integrate the surveillance, signal processing, and decision support technologies in a holistic industrial security system. The IRM DSS functionalities that require an intensive development of new AI techniques are the fusion of information obtained from various sources in real time and the development of optimal decision algorithms that use this fused information. In addition, machine learning techniques, both reinforcement and supervised learning, enhance the selection of preventive and mitigation measures in emergency situations. A preliminary scheme of the AI-based enterprise threat management system for a limestone mine is presented in the next section. The conclusions are provided in sect. 4.

3 Research Issues In Design Of Industrial Risk Management DSS

We will start this section by presenting a formal background to industrial risk modelling, referring to the real-life problems occurring in an opencast mine as a motivating example. However, we claim that the presented models will be useful in solving a broad family of related industrial risk and safety management problems.

We propose a general model of risk measurement as a network of information processing modules, which includes sensors, information fusion, decision support, and automated decision making nodes. Direct signals from threat and risk measurements serve as input to the DSS and risk management and optimization module. The decision algorithms identify potential best-compromise actions and actuators to implement them, and the DSS proposes the outcomes to the decision makers. For the aims of the present study, we assume that the risk model features the following objects:

- multiple threats, natural or anthropogenic,
- two types of information fusion nodes: these capable of fusing information of the same kind from different sensors (simple fusion), and complex fusion nodes capable of processing heterogeneous information,
- endangered humans, whilst dangers can face their health or life,

- artefacts-at-risk, such as machines, vehicles, buildings etc.; threat propagation is depicted as a subgraph of the overall multigraph,
- lower, middle, and top-level decision units, artificial or human,
- rescue teams, robots, actuators (such as extinguishers) that all implement emergency management decisions; for the sake of brevity termed 'responders',
- additional resources that can be used as responders by the enterprise crisis management when necessary.

The model objects and relations between them form a dynamic directed multigraph, with three types of edges that denote information flow, threat propagation and impact, and decision transmission and impact (deontic structure). In the next step, this basic multigraph will be extended with auxiliary structures, such as production line schemes, where a threat imposed on a predecessor causes production interruptions at a subsequent nexus. To make possible application of ML methods and emergency management based on quantitative analysis, the following rules should apply:

- Several information sources reporting the same threat should be adjacent to an information fusion node.
- The information flows are labelled with information credibility coefficients [5].
- There is an information flow path from each information source and each information fusion node to a decision node.
- Each responder is linked by an incoming path to at least one decision node.
- Decision nodes, decisions transferred to lower-level units, rescue teams and actuators, as well as feedback information on their consequences, form a hybrid anticipatory network [7], so that each decision unit is linked to a top-level unit.

A general functional scheme of IRM DSS applicable to a broad class of industries is shown in Fig. 1. This type of diagram will be termed *threat-risk-response map*.

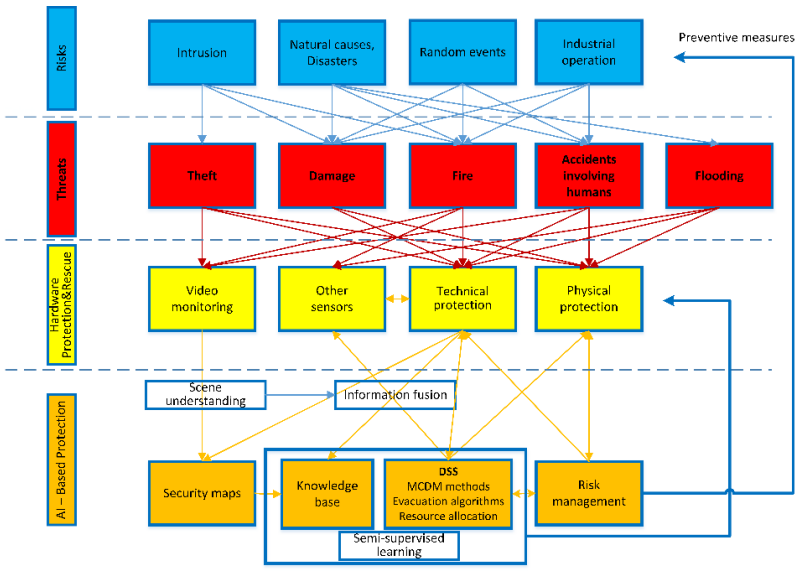


Fig. 1. An example of threats, sensors, information fusion units, and decision making network.

4 Concluding Remarks

In case of emergency, intelligent IRM DSSs will be used sequentially within a semi-supervised ML procedure, where the threat, prior decisions and their outcomes stored in the knowledge base are used to learn current risk mitigation and prevention parameters. The design of an IRM DSS for a limestone mine is a typical system deployment in large industrial area. It showed the benefits of using AI methods such as information fusion, semi-supervised and reinforcement learning, and image understanding to increase risk management capabilities and efficiency. The proposed AI-based DSS design approach can provide viable implementations capable of solving heterogeneous industrial threat management problems in various industries in real time. The IRM-DSS architecture ensures simultaneous optimization of safety management criteria defined in the organization's safety assurance strategy and economic indicators.

Further research on AI-based risk assessment and mitigation in large industrial plants will be focused on the analysis and selection of decision support methods, and new DSS architectures. The specificity of threats and risks will require more penetrative risk modelling and risk response with specific preventive and mitigation actions., If there is a need to manage natural hazards with own forces in large industrial areas, it is necessary to provide both, appropriate instruments and decision-making procedures. We argue that they all can be implemented as a holistic IRM-DSS.

References

1. Chen, D., Liu, Z., Wang, L., Dou, M., Chen, J., Li, H.: Natural disaster monitoring with wireless sensor networks: A case study of data-intensive applications upon low-cost scalable systems, Mobile Netw. Appl. 18(5), 651–663 (2013), doi:10.1007/s11036-013-0456-9.
2. Domdouzis, K.: Artificial-intelligence-based service-oriented architectures (SOAs) for crisis management. In: Handbook of Research on Investigations in Artificial Life Research and Development, pp. 79–95, IGI (2018), https://doi.org/10.4018/978-1-5225-5396-0.ch005.
3. Ibrahim, A.M., Venkat, I., Subramanian, K.G., Khader, A.T., De Wilde, P.: Intelligent Evacuation Management Systems: A Review. ACM Trans. Intell. Syst. Technol. 7, Art. No. 36, p. 27 (2016), https://doi.org/10.1145/2842630.
4. Kerr, C., Phaal, R.: Technology roadmapping: Industrial roots, forgotten history and unknown origins. Technol. Forecast. Soc. Change 155, Art. No. 119967, p.16 (2020).
5. Moyano, F., Fernandez-Gago, C., Lopez, J.: A Conceptual Framework for Trust Models, Trust, Privacy and Security in Digital Business, Lecture Notes in Computer Science 7449, Springer-Verlag, Berlin-Heidelberg, pp. 93-104 (2012).
6. Skulimowski, A.M.J.: Universal intelligence, creativity, and trust in emerging global expert systems. In: L. Rutkowski et al. (eds.), Artificial Intelligence and Soft Computing. ICAISC 2013, Proceedings. Lecture Notes in Artificial Intelligence 7895, Springer-Verlag, Berlin–Heidelberg, pp. 582-592 (2013), https://doi.org/10.1007/978-3-642-38610-7_53.
7. Skulimowski, A.M.J.: Anticipatory Network Models of Multicriteria Decision-Making Processes. Int. J. Systems Sci. 45(1), 39-59 (2014), doi:10.1080/00207721.2012.670308.
8. Skulimowski, A.M.J., Bañuls, V.A.: AI Alignment of Disaster Resilience Management Support Systems. In: Rutkowski L. et al. (eds), Artificial Intelligence and Soft Computing. ICAISC 2021. Lecture Notes in Artificial Intelligence 12855, Springer, pp. 354-366 (2021), https://doi.org/10.1007/978-3-030-87897-9_32.

Algorithm Selection for Population-Based Optimization of Networked Structures

Adam Dziomdziora[1][0000−0002−9667−8689]
and Przemysław Ignaciuk[1][0000−0003−4420−9941]

Lodz University of Technology, Institute of Information Technology
Wolczanska 215, 90-924 Lodz, Poland
adam.dziomdziora@dokt.p.lodz.pl

Abstract. This paper addresses the problem of optimizing goods flow in complex logistic networks influenced by significant demand uncertainty. For that purpose, two population-based algorithms – Biogeography-Based Optimization (BBO) and Particle Swarm Optimization (PSO) – are compared. As considerable computational resources are needed to determine the cost function value, one opts for an algorithm that reaches convergence in the fewest possible steps. The considered fitness function balances the bullwhip effect and transportation cost reduction. For topologies up to 14 nodes, PSO happens to decrease the cost function more than BBO. However, it requires more iterations to reach the optimum. When the interconnection complexity grows, PSO demands larger population size, thus more computations to outperform BBO. For larger topologies, BBO proves more efficient.

Keywords: Population-Based Algorithms · Bullwhip Effect · Logistic Networks · Time-Delay Systems.

1 Introduction

One of the economic areas with rising profitability is logistics, which incorporates manufacturing, transport, and trade [1, 2]. Despite substantial effort to limit its impact, the bullwhip effect (BE) may thwart the process of goods distribution. The BE manifests itself as an intensified variability of demand projected onto resource replenishment decisions. It leads to unnecessary shipments, prolonged delays, and stock accumulation at subsidiary nodes [3], thus should be avoided as early as in the network planning phase. Owing to the significant computational complexity in solving the optimization task, population-based algorithms are considered as a viable alternative to the traditional methods [4]. However, since the cost function calculation in the considered class of problems is demanding, only those methods that permit reaching the optimum in a small number of steps (iterations) are suitable. The purpose of this work is to examine two population-based algorithms: Biogeography-Based Optimization (BBO) and Particle Swarm Optimization (PSO) in optimizing the operational efficiency and preventing the BE in the systems managed by the classical order-up-to inventory policy [5]. The

optimization objective is to shape the interconnection structure, i.e., to decide how intensively a given transportation link (channel) should be used to avoid the BE.

2 Population-Based Optimization

The decisive factor behind cost-efficient operation of a logistic network is an appropriate choice of channel utilization coefficients. The optimization procedure yields a matrix of coefficients reflecting the network structure with reduced propensity for BE formation and smaller transportation costs. Here, BBO and PSO are evaluated as a computational platform for solving the optimization task. BBO has emerged recently as an attractive method for solving complex optimization problems [6], whereas PSO is a mature and well-explored method in diverse applications [7].

2.1 Biogeography-Based Optimization

BBO is an evolutionary, population-based algorithm that derives from the observation of species movement among separate areas called islands. Its power lies in merging examination with exploitation originating from migration. BBO does not require computing fitness function derivatives, yielding an attractive computational platform. The elitism mechanism, missing in PSO, seems to have a crucial impact on the algorithm performance toward cost function reduction.

2.2 Particle Swarm Optimization

PSO is an optimization procedure originating from swarm intelligence with random movement of candidate solutions. It delivers a population-based search procedure where the set of feasible solutions evolves to yield an optimum. Each particle denotes a viable solution to the problem. PSO starts with a swarm - population - that moves through the search space to find the optimal solution. PSO is distinctly different from BBO as it does not utilize crossover and mutation stages. It models the dynamics of candidate solutions through the search space, where each particle moves with a predefined velocity. The experience of every particle and its neighbors governs the swarm movement. Moreover, the swarm communicates to yield attuned parameters for all particles. In an epoch, each particle holds its local best fitness. Also, the global best of the swarm is updated accordingly. The main idea of PSO is to accelerate particles toward their local best and the global best in each iteration.

3 System Model

The examined class of system concentrates on the interaction between two types of actors: external sources – supplying goods to the controlled network and

controlled nodes – generating replenishment signals and serving as intermediate sources for other controlled nodes. The system encompasses N controlled nodes and M external sources, connected by unidirectional links. The system used in experiments is implemented with the methodology introduced in [4]. Here, the simulation size Θ is calculated from the generation size g=10^2, simulation horizon H=10^3 and population size p as:

$$\Theta = g * H * p. \tag{1}$$

4 Numerical Study

In the experiments, the bullwhip effect is quantified through a network indicator (BI) ω, transportation costs Ψ and fitness function FF=Ψ*Ω as in [9]. System results from conducted optimization are grouped in Table 1, where the highlighted ones denote the best minimization. Example topology (Network A) is illustrated in Fig. 1. According to a set of recommended PSO parameters, for the optimal algorithm operation, the population size requires large values, e.g., for a 20-dimension problem, Pendersen recommends p=60. Hence, 20-dimension and greater networks (A and B), resulted in lower costs with BBO. Smaller ones (C and D) achieved better cost reduction with PSO, having p=16 for 11-dimension Network D and p=32 for 14-dimension topology Network C. Also, one should notice that enlarging the population size significantly prolongs the overall simulation time. On the other hand, PSO could obtain better results with appropriately adjusted population size, though it requires further investigation.

Table 1: Population size dependence for networks with p=16 $\vee$ 32.

(a) Transportation costs

***p*=32**	Nodes	Ψ (€)	Ψ_{PSO}	Ψ_{BBO}
A	N=25, M=6	1.39 x 10^8	1.27 x 10^7	**9.14 x 10^6**
B	N=15, M=5	1.45 x 10^7	3.7 x 10^6	**3.01 x 10^6**
C	N=12, M=2	8.77 x 10^7	**2.27 x 10^6**	2.87 x 10^6
D	N=9, M=2	1.94 x 10^7	**1.22 x 10^6**	1.56 x 10^6
***p*=16**	Nodes	Ψ (€)	Ψ_{PSO}	Ψ_{BBO}
A	N=25, M=6	1.46 x 10^8	1.43 x 10^7	**9.31 x 10^6**
B	N=15, M=5	1.45 x 10^7	3.7 x 10^6	**3.01 x 10^6**
C	N=12, M=2	5.86 x 10^7	3.76 x 10^6	**3.38 x 10^6**
D	N=9, M=2	1.88 x 10^7	**1.73 x 10^6**	1.8 x 10^6

(b) BI

	Ω	Ω_{PSO}	Ω_{BBO}
A	14.99	4.74	**4.22**
B	7.36	3.7	**3.17**
C	18.07	**3.08**	3.17
D	8.21	3.12	**2.99**
	Ω	Ω_{PSO}	Ω_{BBO}
A	16.31	4.94	**4.29**
B	4.89	3.42	**3.2**
C	12.4	3.71	**3.16**
D	7.27	2.81	**2.68**

5 Conclusions

In 14-node and smaller networks, PSO yields more profound cost reduction than BBO. However, it performs better only after the population size is increased

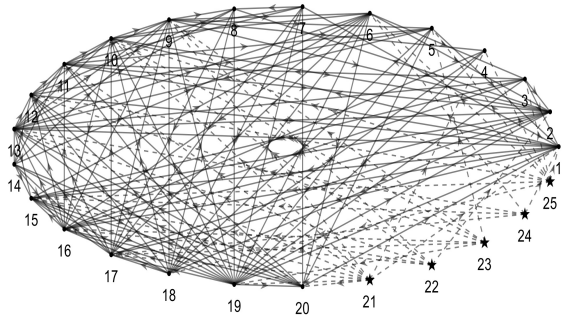


Fig. 1: Example network structure. Stars denote the exterior suppliers (M=5), and circles illustrate controlled nodes (N=20).

from 16 to 32 individuals, thus significantly extending the simulation time. In 15-node and bigger topologies, BBO outperforms PSO and converges faster. In the future studies, BBO will be compared with PSO in the network optimization under a much larger population size setting, e.g., p>100.

Acknowledgment

This work has been performed in the framework of a project "Robust control solutions for multi-channel networked flows" no. 2021/41/B/ST7/00108 financed by the National Science Centre, Poland. This work has been completed while the first author was the Doctoral Candidate in the Interdisciplinary Doctoral School.

References

1. Sumukadas, N.: Are you ready for your next product recall crisis? Lessons from operations and supply chain management. Bus. Horiz. **64**(2), 211–221 (2021).
2. Speranza, M.: Trends in transportation and logistics. Eur. J. Oper. Res. **264**(3), 830–836 (2018).
3. Lee, H., Padmanabhan V., and Whang S.: The Bullwhip Effect in Supply Chains. Sloan Manage. Rev. **38**(3) (1997), 93–102 (1997).
4. Ignaciuk, P. and Wieczorek, L.: Continuous Genetic Algorithms in the Optimization of Logistic Networks: Applicability Assessment and Tuning. Appl. Sci. **10**(21), 7851 (2020).
5. Axsäter, S.: Inventory control. Springer International Publishing (2015).
6. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008).
7. Pedersen, M.: Good parameters for particle swarm optimization. Technical report, Hvass Laboratories, www.hvass-labs.org (2010).
8. Simon, D.: Evolutionary optimization algorithms. John Wiley & Sons (2013).
9. Ignaciuk, P. and Dziomdziora, A.: Quantifying the Bullwhip Effect in Networked Structures with Non-trivial Topologies. In: ACM International Conference Proceeding Series, ACM, New York, USA, 62–66 (2020).

Dominance-based Rough Set Approach to Bank Customer Satisfaction Analysis

Marcin Szeląg[1] and Roman Słowiński[1,2]

[1] Institute of Computing Science, Poznań University of Technology, Poznań, Poland
[2] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
{mszelag,rslowinski}@cs.put.poznan.pl

Abstract. We present an application of the Dominance-based Rough Set Approach (DRSA) to analysis of bank data concerning customer satisfaction. The analysis is conducted using two new applications – RuLeStudio and RuleVisualization. The first one is designed to experiment with different rule models, depending on chosen settings. The second one is used for visualization and in-depth examination of the constructed model. Our analysis gives insight into the data in terms of monotonic decision rules which describe loyal customers and the ones who ceased cooperation with the bank. Such analysis is in vain of explainable AI, aiming to obtain a transparent decision model, that can be understood by decision makers. We also compare predictive performance of our approach with some well-known machine learning methods.

Keywords: Dominance-based Rough Set Approach · Ordinal classification · Monotonic relationships · Decision rules · Customer Satisfaction.

1 Introduction

Rough set theory (RST) was introduced by Zdzisław Pawlak [5]. Since then, different extensions and applications have been proposed. An important direction of research, initiated by Greco, Matarazzo, and Słowiński, concerns adaptation of RST to multicriteria decision aiding. They proposed the Dominance-based Rough Set Approach (DRSA) [4], which employs dominance instead of indiscernibility relation among objects in the definition of rough approximations, and builds decision models in terms of monotonic $if \ldots, then \ldots$ decision rules. DRSA is able to take into account monotonic relationships present in data between condition and decision attributes. Rule models are considered to be both transparent for a user, and useful for explanation of suggested decisions, which is an important aspect of AI methods, apart from sole predictive performance.

In this study, we show an application of DRSA to analysis of a bank customer data. Employing decision rules, we wish to present readable patterns of customers who left the bank. We use Variable Consistency DRSA (VC-DRSA) [2] and introduce a new rule classifier. We also use a new software designed to learn, explore and apply decision rules.

In Section 2, we describe the methodology. Section 3 presents the analysis of public domain data obtained from a bank. The last Section 4 groups conclusions.

2 Methodological Background

We employ VC-DRSA with cost-type consistency measure ϵ [2], used to calculate lower approximations of unions of ordered decision classes, and to induce decision rules from these approximations. This involves threshold $\theta_X \in [0, 1]$ to be defined by the user for each upward/downward union $X \subseteq U$, where U is a universe of (learning) objects. Then, the lower approximation of X is composed of objects whose consistency (measured by ϵ) is not worse than θ_X. The rules are induced using *VC-DomLEM* algorithm [3], and later, those with confidence ≤ 0.5 are removed. They explain observed decisions, and can classify any new object.

In DRSA, classification of an object based on matching rules can be done in different ways (see, e.g., [1]). We propose a *mode classifier* being able to resolve conflicting class assignments. It is implemented in RuLeStudio[3]. Let consider the set of objects described in terms of two gain-type criteria g_1, g_2 shown in Fig. 1. Let denote class i by Cl_i. Object z to be classified (red cross) is covered by rules

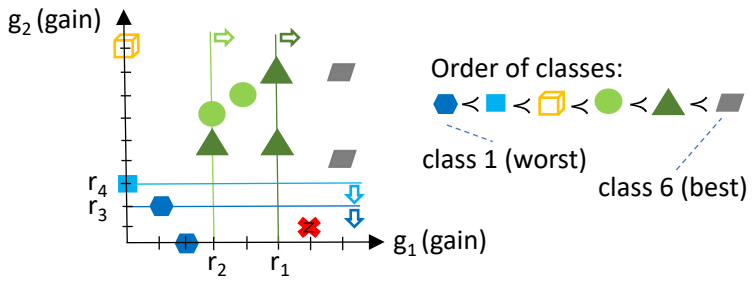


Fig. 1. Exemplary set of objects for illustration of the *mode classifier*

r_1, r_2, r_3, r_4, suggesting, respectively: "at least Cl_5", "at least Cl_4", "at most Cl_1", and "at most Cl_2". Then: (i) *upward intersection* is "at least Cl_5", (ii) the most prudent upward class is Cl_5, (iii) *downward intersection* is "at most Cl_1", (iv) the most prudent downward class is Cl_1, (v) *mode* of the two classes is computed. Observe that r_1 covers 2 objects from Cl_5, and r_2 covers 1 additional object from Cl_5. Then, Cl_5 is supported by 3 objects. Moreover, r_3 covers 2 objects from Cl_1, and r_4 covers no additional object from Cl_1. Then, 2 objects support Cl_1. Consequently, Cl_5 is returned by the classifier (more frequent class).

If no rule matches z, one can suggest a majority class (optimizing classification accuracy) or median class (optimizing mean absolute error).

3 Case Study of Bank Customer Satisfaction

We analyze the *churn* data set publicly available at kaggle.com[4], featuring 10 condition attributes, incl. 4 continuous ones. To build a balanced universe of

[3] www.cs.put.poznan.pl/mszelag/Software/RuLeStudio/RuLeStudio.html
[4] https://www.kaggle.com/mathchi/churn-for-bank-customers

objects U, we drew 2000 churning customers ($Exited = 1$) and 2000 loyal customers ($Exited = 0$)[5]. In this study, we compare our method (ϵ-VC-DRSA + mode classifier; $Exited = 0$ as default decision) with three ML classifiers available in WEKA[6] (with default parameters): SVM (SMO) with polynomial kernel, C4.5 (J48) tree classifier, and naive Bayes (NaiveBayes) classifier. We estimate predictive performance using classification accuracy.

We considered the remarks at kaggle.com, WEKA's histograms, and trial-and-error assessment in RuLeStudio to assign attribute preference orders as follows: *CreditScore* – gain (after kaggle.com), *Geography* – none (nominal attribute), *Gender* – none (nominal attribute), *Age* – cost (distribution for class $Exited = 1$ shifted to the right), *Tenure* – cost (verified in RuLeStudio), *Balance* – gain (kaggle.com), *NumOfProducts* – we duplicated this attribute and assigned type gain to the first clone, and type cost to the second one (the histogram shows prevalence of loyal customers when $NumOfProducts = 2$, and the opposite otherwise), *HasCrCard* – none (nominal attribute), *IsActiveMember* – gain (kaggle.com) *EstimatedSalary* – gain (kaggle.com). For the decision attribute *Exited*, label 0 was more preferred than 1 (bank's viewpoint).

In our study, unions of classes boil down to single classes – characterized by decisions $Exited = 0$ and $Exited = 1$. We assumed a common threshold θ_X for both classes. Using cross-validation in RuLeStudio, we tested thresholds 0, 0.01, 0.02, and 0.05, choosing value 0.01. Note that for $\theta_X = 0$ (classical DRSA), the quality of classification was 0.68775, while for $\theta_X = 0.01$ it increased to 0.996.

Table 1 presents comparison of average classification accuracy from 3 independent runs of 10-fold cross-validation. One can see that our method performed better than SVM, slightly better than naive Bayes, and slightly worse than C4.5.

Next, we analyzed the models trained on all 4000 objects. Reclassification accuracy was: SVM 70.225%, naive Bayes 72.25%, C4.5 85.525%, our approach 83.825% (2nd best). C4.5 tree size was equal to 320 with 164 leaves. The tree had many long paths which were hard to understand and did not respect the above preference orders. When transformed to 164 rules, even after aggregating redundant conditions (e.g., Age $\leq$ 41 and Age $\leq$ 37 resulted in Age $\leq$ 37), average rule length was 7.81 and average rule support was 24.39. The model learned by ϵ-VC-DRSA contained 770 rules. We explored them in RuleVisualization[7]. Our observations: (i) on avg. 5.91 conditions per rule – much better than C4.5; (ii) avg. rule support 34.1 – again much better than C4.5; (iii) top

Table 1. Comparison of average classification accuracy in 3 × 10-fold cross-validation

method	**ϵ-VC-DRSA+mode**	SVM	C4.5	naive Bayes
avg. accuracy	73.25	69.91	75.18	71.87
rank	2	4	1	3

[5] http://www.cs.put.poznan.pl/mszelag/Research/bank-churn
[6] https://www.cs.waikato.ac.nz/∼ml/weka; used version: 3.8.6
[7] www.cs.put.poznan.pl/mszelag/Software/RuleVisualization/RuleVisualization.html

attributes present in rules are: Geography (in 76.2% of rules), Age (in 74.9% of rules), EstimatedSalary (in 59.9% of rules), CreditScore (in 58.7% of rules); (iv) most often co-occurence of attributes concerns Geography and Age; (v) the two strongest rules concern decision Exited ≥ 1 and are supported by 279 and 221 objects. Fig. 2 shows top rules for customers who left the bank (Support ≥ 100, Confidence ≥ 0.95). Remark that NumOfProducts ≥ 3 is often related to churn.

ID	Conditions	Decision	Epsilon	Support
516	Age ≥ **49**, IsActiveMember ≤ **0**, NumOfProducts_g ≤ **1**, CreditScore ≤ **788**	Exited ≥ **1**	0.006	279
420	NumOfProducts_c ≥ **3**, Age ≥ **38**	Exited ≥ **1**	0.002	221
506	Age ≥ **50**, IsActiveMember ≤ **0**, CreditScore ≤ **646**, HasCrCard = **1**	Exited ≥ **1**	0.004	141
422	NumOfProducts_c ≥ **3**, Geography = **France**, Age ≥ **31**	Exited ≥ **1**	0.001	106
517	Age ≥ **49**, IsActiveMember ≤ **0**, Geography = **Germany**, CreditScore ≤ **664**	Exited ≥ **1**	0.003	104
427	NumOfProducts_c ≥ **3**, Gender = **Male**, Age ≥ **35**	Exited ≥ **1**	0.001	101
421	NumOfProducts_c ≥ **3**, CreditScore ≤ **657**, Gender = **Female**	Exited ≥ **1**	0.001	100

Fig. 2. Top rules describing customers who ended cooperation with the bank

4 Conclusions

In this paper, we analysed customer satisfaction data from a bank using Variable Consistency Dominance-based Rough Set Approach, and some reference machine learning methods. We employed two new programs suitable for this task: RuLeStudio and RuleVisualization. Moreover, we proposed a new rule classification strategy (mode classifier), implemented in RuLeStudio. The results obtained using our approach are competitive with respect to average classification accuracy, but even more important, the induced rule model gives a clear insight into the problem, helping the bank to improve long-term customer relationships.

Acknowledgements This research was supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

1. Błaszczyński, J., Greco, S., Słowiński, R.: Multi-criteria classification – a new scheme for application of dominance-based decision rules. European Journal of Operational Research **181**(3), 1030–1044 (2007)
2. Błaszczyński, J., Greco, S., Słowiński, R., Szeląg, M.: Monotonic variable consistency rough set approaches. Int. J. of Approx. Reasoning **50**(7), 979–999 (2009)
3. Błaszczyński, J., Słowiński, R., Szeląg, M.: Sequential covering rule induction algorithm for variable consistency rough set approaches. Information Sciences **181**, 987–1002 (2011). https://doi.org/10.1016/j.ins.2010.10.030
4. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research **129**(1), 1–47 (2001)
5. Pawlak, Z.: Rough sets. Int. J. of Inf. & Computer Sciences **11**, 341–356 (1982)

Double Genetic Algorithm for Vehicle Routing Problem

Joanna Ochelska-Mierzejewska[1][0000−0002−9295−3962]

Institute of Information Technology, Lodz University of Technology
joanna.ochelska-mierzejewska@p.lodz.pl

Abstract. A Vehicle Routing Problem (VRP) is a combinatorial problem that seeks the optimal set of routes for a fleet of vehicles delivering a set of shipments to customers. This problem is very important for the logistics industry. Optimizing delivery routes can cut costs by up to 25%. The following paper proposes an algorithm that allows calculating routes that are close to the optimal ones for routes consisting of 100 destination points. The proposed algorithm is a combination of two genetic algorithms. The first part is an external algorithm to assign target points to each vehicle. Then, for each vehicle, an internal genetic algorithm is applied, which, within a single vehicle, determines the shortest route between the parcels to be delivered.

Keywords: Vehicle Routing Problem · Genetic Algorithm.

1 Introduction

One of the most frequently chosen optimization transport problems is the Vehicle Routing Problem (VRP) which uses more than one vehicle which has the beginning and the end of their route at the same point. For the first time, the solution algorithm of this problem was described by Dantzig and Ramser in 1959 [1].

This issue can easily be translated into a real problem which can be defined as a company that provides its customers a certain amount of homogeneous goods every day. The main company reason is to deliver the assumed amount of goods to each of the customers, meeting the following conditions: each customer receives exactly the amount of goods they need, the quantity of well-delivered by each vehicle may not exceed its capacity, each customer can be visited only once, the sum of the route costs of all vehicles used should be as low as possible [2].

This aim of the paper is to analyze the vehicle routing problem using a double genetic algorithm (internal and external) and to compare the results of changing the number of points and the number of goods to be collected from reloading points.

The paper is divided into 3 sections, each of which describes a different topic. After the introduction, in section 2 there is a description of the genetic algorithm implemented for VRP. The proposed algorithm is a combination of two genetic algorithms. The first part is an external algorithm to assign target points to each

vehicle. Then, for each vehicle, an internal genetic algorithm is called, which determines the shortest route between the shipments to be delivered within a single vehicle. The most important is section 3, where there are described experiments and discussion of results..

2 Genetic Algorithm for Vehicle Routing Problem

Work on evolutionary systems, which genetic algorithms are part of, began in the 1950s. Based on evolution strategies, a group of researchers from the University of Michigan developed in the 1960s and 1970s the idea of genetic algorithms (GA) [3]. In contrast to research on evolutionary programming, the general idea was to explore adaptation based on nature observations and a way to incorporate that knowledge in machine processing.

Selection ensures that the strongest individuals understood as better solutions, have a larger probability to produce offspring [4]. Offspring, on the other hand, is generated by reproduction, which takes the form of crossing bits of one solution with bits of another one [6]. Finally, the randomized mutation is introduced, usually in form of swapping two bits, or negating just one of them in an individual in the whole population [5].

The classical approach to Genetic Algorithms is not valid for sequential problems such as Vehicle Routing Problem. The main problem is that a single solution cannot be represented as a set of bits. The proper encoding of a solution is a string of integer numbers, representing vertices of a problem-space graph [7].

We divide the issues of the VRP problem into the *external* algorithm were optimizing the placement of destination points for delivery vehicles and the *internal* algorithm was optimizing the route of the delivery vehicle.

The internal algorithm consists of three steps:

1. *Coding*. The adaptation of genomes of solutions to the VRP problem is being implemented as a list of delivery points for a given vehicle.
2. *Crossover*. Then the elite selection method is used with the adaptation function. Then the chosen ones are matched in pairs and crossed. At the very end, the process of mutation takes place.
3. *Function of adaptation*. The following rule applies here: the smaller the value of the fitness function, the less adaptation. It computes for everyone's destination point the distance between the next point on the list. Next, it is checked whether the current duration of the vendor's work is added to the distance covered in a given iteration. If the condition is true, the huge number is returned as an adaptation. After visiting all points on the road, you are returned is the value of the fitness function, which is the sum of the distance traveled from the starting point to the first point, distance traveled between all points and the distance from the last point to the starting point.

The external algorithm also consists of three steps:

1. *Coding*. The adaptation of genomes of solutions to the VRP problem creates a route list for a given vehicle.

2. *Crossover*. Then follows the elite selection method is used with an adaptation function. The selected route genomes are then paired and crossed. In the end, the process of mutating is realized.
3. *Function of adaptation*. The following rule applies here: the smaller the value of the fitness function, the less adaptation. In the beginning, a fleet of suppliers is created equal to the number of possible routes. Then it is checked whether any of the vehicles have failed contain loads exceeding the vehicle's capacity by weight. If such a case does not occur, the internal algorithm described is triggered above. Otherwise, the fleet returns a large number indicative of poor adjustment.

The above algorithms work in loops with a given number of iterations. In the end, the adaptation of the best supplier fleet is returned.

3 Experiments

The experiments were performed in the study, to determine the optimal algorithm parameter values. It was tested on and verified on Solomon benchmark data sets: population size of the external algorithm and population size of the internal algorithm.

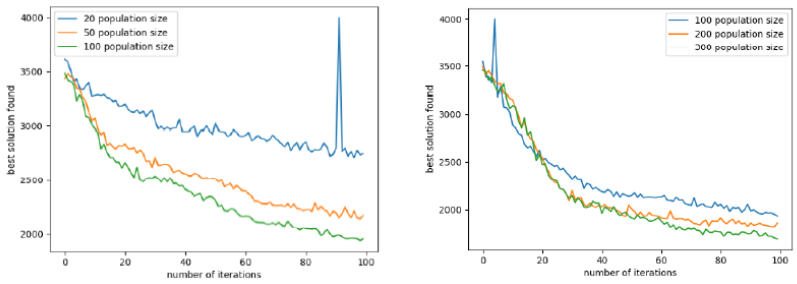


Fig. 1. Effect of the *external* algorithm's population size on the result.

There was a clear correlation between the size of the population and the quality of the results (Fig. 1). However, this parameter is also linearly correlated with the execution time algorithm. To optimize the execution time, the algorithm was determined at a value of 200 for the external algorithm population because above these values, only a slight increase in the quality of the results was observed.

There was a clear correlation between the population size of the internal algorithm with the quality of the results, however, this parameter is also linearly correlated with the execution time of the algorithm (Fig. 2). The above-mentioned correlation was also found parameter to the maximum achievable result. This relationship is because the higher population of the internal algorithm allows for finding optimal solutions for cars with a larger number of assignments points.

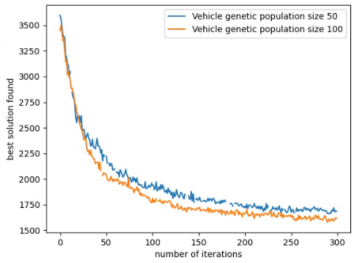


Fig. 2. Influence of the *internal* algorithm's population size on the result.

4 Summary

In experiments there were found the suitable parameters' values for *external* and *interneal* genetic algorithm. The results showed that the optimal parameters for the presented algorithm are: external algorithm population is equal to 200, internal algorithm population is equal to 100. Next, these values were used to check the correctness of results of benchmark data – the result are promising and worth further research. These experiments showed that the double genetic algorithm can be used to solve the VRP problem.

References

1. Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem, Management Science **6(1)**, (1959)
2. Ji, X., Yong, X.: Application of Genetic Algorithm in Logistics Path Optimization. Academic Journal of Computing & Information Science, **2**, pp. 155-161 (2019)
3. Mitchell M.: An Introduction to Genetic Algorithms, MIT Press (1998)
4. Ochelska-Mierzejewska, J., Poniszewska-Marańda, A., Marańda, W.: Selected Genetic Algorithms for Vehicle Routing Problem Solving. Electronics **10(24)**:3147 (2021) https://doi.org/10.3390/electronics10243147
5. Ochelska-Mierzejewska, J., Zakrzewski, P.: Comparison the Genetic Algorithm and Selected Heuristics for the Vehicle Routing Problem with Capacity Limitation. In: Kryvinska N., Poniszewska-Marańda A. (eds) Developments in Information & Knowledge Management for Business Applications. Studies in Systems, Decision and Control, **377**, Springer, Cham (2022) https://doi.org/10.1007/978-3-030-77916-0_10
6. Saxena, R., Jain, M., Malhotra, K., Vasa, K.D.: An Optimized OpenMP-Based Genetic Algorithm Solution to Vehicle Routing Problem. In: Elci, A., Sa, P., Modi, C., Olague, G., Sahoo, M., Bakshi, S. (Eds.): Smart Computing Paradigms: New Progresses and Challenges. Advances in Intelligent Systems and Computing, **767**, Springer (2020)
7. Toth, P., Vigo, D.: The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia (2001)

Explaining the Shallow Network Trained to Classify EEG Signals by LRP and Sensitivity Analysis

Martyna Poziomska[1], Alan Drozd[2], Urszula Malinowska[1[0000-0001-8358-6024]], Jacek Rogala[2[0000-0001-5298-920X]], and Jarosław Żygierewicz[1[0000-0002-7536-0735]]

[1] Faculty of Physics, University of Warsaw, Pasteura 5, 02-093, Warsaw, Poland
[2] Nencki Institute of Experimental Biology, Polish Academy of Science, Pasteura 3, 02-093, Warsaw, Poland
m.poziomska@student.uw.edu.pl

Abstract. Explanation of EEG classification made by a neural network is still a challenging problem. Therefore, we test two approaches based on sensitivity analysis and Layer-wise relevance propagation. We show that the explanations obtained by these methods are consistent only to a certain degree, as each focuses on different aspects. Sensitivity analysis exposes which changes in the data would increase the probability of a given class, while LRP is more related to the values of specific features in the data. The additional difficulty is ensuring that the indicated features are indeed plausible for contrasting the conditions. Here we used the spectral power distribution across channels in the experimental conditions as a sanity check.

Keywords: Explainable Machine Learning, EEG Classification.

1 Introduction

Neural networks models are often treated as black-boxes, i.e., trained to solve a classification problem. They are effective at assigning class labels to input data. However, in many practical cases, it is important to understand the essential input features used by the model and whether they are reasonable in the given problem or the model uses some artifacts present in the training set. This contribution presents the attempts to assess the importance of the input EEG signal features using two methods.

2 Data

Data come from 87 participants who took part in a delayed match-to-sample experiment. There were two experimental conditions requiring (ATT) or not (CON) retention of an object in the visual memory. The signals used to train the network were 5s long fragments encompassing the delay phase, recorded with 19 electrodes from the 10-20 system, sampled at 400 Hz, and filtered in the 0.5-45 Hz frequency band. In total, there were 10937 CON and 8366 ATT trials.

3 Methods

We used the Shallow-ConvNet architecture as proposed in [1]. In a 3-fold cross-validation (CV) scheme, we trained it to classify trials as ATT or CON, using binary cross-entropy loss and AdamW optimizer. The importance of the signal features utilized by the model was estimated by two approaches.

First, we conducted a sensitivity analysis. It relied on evaluating the gradient of class ATT probability p over the perturbation parameters related to frequency-band power in a given EEG channel. The obtained gradients were averaged across 3 CV folds and 5 random initializations.

The second approach evaluated the relevance of the periods and filters in the first and second Conv layers using Layer-wise Relevance Propagation (LRP) [2]. Filters in the first Conv layer can be interpreted as FIR filters; thus, we can straightforwardly compute their transmittance $H_F(f)$ to obtain their frequency characteristics. We choose the most important filter F^* for each electrode e based on the mean absolute value of the relevance $R_{2,F,e}$ computed for each electrode and each filter F in the second layer of the model. The product of the absolute value of the transmittance $H_{F^*,e}(f)$, the spectral power of the signal (only for ATT) $S_e(f)$, the normalized[1] relative filter weight from the second layer $\hat{R}_{2,F^*,e}$ and the normalized relative filter weight from the firtst layer $\hat{R}_{1,e}$ yields the importance of a given frequency band ($f \pm 2$Hz) at a given electrode $I_e(f)$:

$$I_e(f) = \sum_{f_i=f-2Hz}^{f+2Hz} \left| H_{F^*,e}(f_i) \right| \cdot \mathrm{S}_e(f_i) \cdot \hat{R}_{2,F^*,e} \cdot \hat{R}_{1,e} \quad (1)$$

Additionally, we evaluated the average power spectral density in the two experimental conditions by periodogram. These spectra served as a reference for interpreting the results of features indicated by sensitivity analysis and LRP.

4 Results

The models trained in 3 CV and 5 random initializations achieved Matthews correlation coefficient of 0.22±0.02.

The average importance of frequency bands and channels according to sensitivity analysis is shown in Fig. 1a. Positive index values indicate that an increase of power in a given frequency band at a given electrode increases the probability of the input trial being of class ATT. We note a group of frontal (F7, Fz, F4) and temporal (T3, T4) electrodes showing that an increase of theta and alpha power in these regions suggests the trial being of ATT class. In contrast, an increase of theta and alpha power in parieto-temporal and occipital regions indicates the trial being of CON class.

The analysis of the most relevant filters indicated by the LRP method is shown in Fig. 1b. Here, we notice that the electrodes T6 and Pz are especially prominent at alpha and theta frequency bands. But in contrast to the sensitivity analysis, the frontal regions are not highly relevant.

[1] Divided by the max value

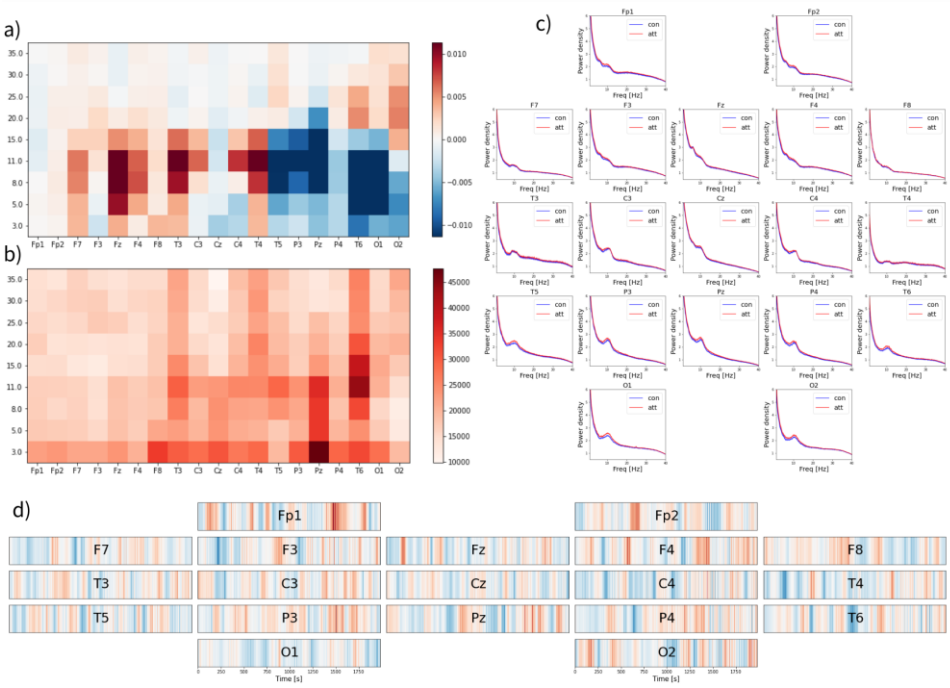


Fig. 1. a) Results of the sensitivity analysis; b) Importance of channels and frequency bands in the LRP analysis; c) Average spectral power, with standard error of measurement marked, for ACC and CON conditions; d) time course of relevance in the EEG channels.

We show average spectra for ATT and CON experimental conditions (Fig. 1c.) as a reference for the above-described results. Again, we observe an alpha rhythm peak, especially prominent in the central and parietal regions. Moreover, a closer look reveals that the power for ATT in the alpha peak is higher than for the CON class. Additionally, we notice a peak corresponding to theta rhythm in the frontal regions.

The relevance propagated to the input signals enables us to estimate the importance of various time periods at different channels. We visualize it in Fig. 1d.

5 Conclusion

The current study demonstrates that the sensitivity analysis and LRP based approaches are promising in explaining the neural network models trained to EEG classification. Still, in contrast to applying these methods to explaining models in the context of image classification, it is much more challenging to demonstrate the correctness of the indication of the critical features.

References

1. Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., Ball, T.: Deep learning with convolutional neural networks for EEG decoding and visualization. Hum. Brain Mapp. 38, 5391–5420 (2017) DOI: 10.1002/hbm.23730
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE, vol. 10, no. 7, Art. no. e0130140 (2015)

Fusion of Expert Information in Content-Based Multimedia Retrieval with Group Decision Support

Andrzej M.J. Skulimowski [0000-0003-0646-2858] 1,2
[1] AGH University of Science and Technology, Chair of Automatic Control and Robotics, Decision Science Laboratory, Mickiewicza ave. 30, 30-059 Kraków, Poland
[2] International Centre for Decision Sciences and Forecasting, Progress & Business Foundation, J. Lea street 12B, 30-048 Kraków, Poland
ams@agh.edu.pl

Abstract. This paper proposes a new approach to fuse recommendations of experts involved in multicriteria decision support procedures. Recommendations are formulated by experts independently as elements of four predefined reference sets. The decision situation corresponds either to the scheme 'one decision maker – multiple recommending experts' (group decision support) or 'multiple decision makers – multiple experts' (group decision making and support). We define the internal, mutual, and plausibility inconsistencies and propose a procedure to verify consistency, regularize the reference values, and fuse them. An expert's trust coefficient decrease when a reference value recommended by this expert is modified during the regularization process. Thus, the regularization outcomes can be assessed ex ante with respect to the total losses of experts' credibility and to the quality of the resulting reference set. The decision process will be illustrated by an example of content-based multimedia retrieval from the AI-based knowledge repository developed within a recent research project.

Keywords: Multicriteria decision making, Content-based information retrieval, Reference set method, Multiple reference points, Group decision support

1 Introduction

In this article we consider the situation, where multiple experts involved in decision support provide independent hints to one or more decision makers. These seek a best compromise selection of multimedia learning material to be retrieved from an AI-based learning platform (AILP). Experts' recommendations are formulated as reference points of different kind. By definition, the reference points are distinguished elements of the criteria space $E:=\mathbf{R}^N$, which are assigned a special meaning to one or more decision maker(s) in the multicriteria optimization problem

$$[F:=(F_1,\ldots,F_N):U\rightarrow E]\rightarrow min(\theta), \tag{1}$$

where θ is an ordering cone in E. So defined reference points can be used as additional preference information in the extended Topsis [2], bipolar [7], reference set (Ref-Set, [4]), or safety principle method. The multicriteria decision making (MCDM) methods based on reference points have been widely used to solve real-life compromise

decision selection in problem (1) such as industrial design, staff recruitment, autonomous decision modeling or preference-based filtering in content based image retrieval (CBIR, [3],[6]). The latter area constitutes the background for content-based multimedia retrieval from the knowledge repository [8], where our approach is dedicated to.

The expert decision recommendation procedure with multiple reference points consists in defining several – usually three or more - classes of reference points, while the elements of each class are assumed to be characterized by the same utility value. Below we recall the definitions of four basic classes of reference points [4] and link them to the multimedia retrieval problem:

A_0 - *Bounds of optimality* - the values of A_0 correspond to the alternatives with properties that exceed actual needs, such as too advanced or too extensive courses.

A_1 - *Target points* - the elements of the criteria space that model ideal but usually non-admissible solutions. If none of them can be selected, the decision-maker should try to choose the learning material with parameters as close as possible to the coordinates of any of elements of A_1. These points are also termed ideal aspiration levels.

A_2 - *Status-quo solutions* - attainable criteria values which should be outperformed during the decision-making process, such as parameters of the multimedia learning material already available at the pre-decision stage without further search in the AILP.

A_3 - *Anti-ideal reference points* - express the wrong choice, for example failed learning material selection in the past. Therefore, the set A_3 should be avoided during the decision-making by choosing a solution most distant to decisions represented by A_3.

Reference points can be used to estimate an underlying utility (or value) function v according to the assumption that the elements of A_i are assigned the same deterministic utility value $v(A_i)=\alpha_i>0$ such that $\alpha_i<\alpha_j$ when $j<i$. So, v is to be maximized on U as a scoring function for the problem (1). The background of the multiple reference points (MREF) and reference set methods (RefSet) has been given in [1] and [4], while their extensions related to the CBIR problem are provided in [3] and [6]. In these decision support approaches, neighboring classes of reference points are pairwise aggregated to form ,two subsets of the criteria space, one to be approached and the other to be avoided. This paper is devoted to the study of reference points consistency. We provide the rules and operations to convert inconsistent sets of reference points into the structure that complies with the utility modelling principles and to fuse them.

2 The Consistency of Expert Recommendations

The utility function estimation from expert judgments requires consistency in assigning utility values to reference points. These must comply with the monotonicity of the utility function v. Furthermore, their situation with respect to the attainable set $F(U)$ in E must conform to the common-sense meaning of reference points presented in the previous section. This additional property will be termed here consistency with the problem (1). The multiple reference points approach presumes that all elements of the class A_i correspond to the same value of the estimated utility function. We will call this property *internal consistency* of the class A_i.

Definition 1. *The set of reference points A_i is internally consistent iff*

$$\forall q_1, q_2 \in A_i : q_1 \text{ and } q_2 \text{ are noncomparable with respect to } \theta. \quad (2)$$

Moreover, each one of the sets A_i, $0 \le i \le K$, where K+1 is the number of classes of reference points, should be well-defined with respect to all other sets of reference points. This requirement will be formulated as an assumption that each element of A_j should be dominated by an element of A_{j+1}, for $0 \le j \le K$-1, i.e.

$$\forall x \in A_j \ \exists y \in A_{j+1} : \ x \preceq_\theta y. \quad (3)$$

To obtain the desired properties of the level sets of ν we impose an additional condition (4), symmetric to (3), which allowed us [4] to define the *mutual consistency*,

$$\forall x \in A_{j+1} \ \exists y \in A_j : x \preceq_\theta y. \quad (4)$$

Definition 2. *The reference classes A_j and A_{j+1} that satisfy the conditions (3)-(4) will be termed mutually consistent.* ■

Along with consistency, rationality is a fundamental property of multicriteria decision-making methods that should be verified in the first order of importance.

Definition 3. *A multicriteria decision making procedure is termed rational if its ultimate solution is nondominated with resppect to the order defined in* (1). ■

Due to frequent occurrence of inconsistent hints, specifically in the context of of multimedia learning material selection, checking and correcting mutual consistency turned out to be an essential part of recommendation-based decision-making processes involving the AILP users [5],[8]. This is justified by the following fact:

Theorem 1. *If all reference classes A_i for the problem (1) are both internally and mutually consistent, then the solution process described as RefSet in* [4] *is rational.* ■

After estimating the attainable set $F(U)$ in (1) it may happen that the actual situation of reference points differs from that required by Defs. 1-3 above. Then, it is necessary to re-formulate experts' judgments according to the general rule that the rationality of the compromise solution is superior to the intuitive interpretation of reference points. This is the aim of the regularization procedure that uses a.o. reference point averaging, re-defining, splitting and merging the reference classes. In the next section, this procedure is illustrated with an example referring to multimedia course selection.

3 An Example of Content-Based Multimedia Course Selection

As an application example of our approach, we present a procedure of recommending videos, massive online courses (MOOCs), and other multimedia courses to the users of the AILP developed within the recent Horizon 2020 research project [8]. Instructors are principal recommending experts, while students are the decision makers who select the learning material according to the criteria G related to the achievement of learning goals, learning efficiency and comfort. Usually, the criteria G are expressed as preference-preserving functions of certain machine-measurable pre-criteria F.

Example 1. Assume that two instructors I_1 and I_2 with the same trust coefficients 0.8 recommend multimedia courses according to the qualitative criteria $G=(G_1,G_2)$, where G_1 describes the correctness and clarity of presentation, G_2 - the presence of specific adequate real-life cases. According to (1), G is to be minimized, which means that its numerical values should be interpreted as deviations from a certain ideal course. There are three directly measurable criteria $F=(F_1,F_2,F_3)$, with F_1 - the coverage of the obligatory and auxiliary stuff (in %), to be determined by text mining from the course annotations, F_2 – the average quality of graphics and videos – to be determined by checking the resolution and provenience of images and videos, and F_3 is the availability of supplementary material normalized on the scale [0,1]. The dependence of G on F is disclosed initially in an explicit form for the sets A_i only. However, it can be learned with adaptive regression techniques from earlier recommendation-selection processes that involved other experts and users, i.e. G is represented as $G=\varphi^{\circ}F$. Current recommendations update the hitherto determined coefficients of φ. The learning goals can be made explicit as a query that defines a preliminary selection of the learning content to be considered in further recommendation and multimedia selection process. Consequently, this query corresponds to the constraints U that in this example consist of three nondominated courses such that $G(F((U))=\{(2,4),(3,3),(4,2)\}$. The reference points from A_1 and A_2 are provided in Table 1 as vectors with the corresponding values of G and F. Additionally, we assume that $A_0=\{(0,0)\}$ and $A_3=\{(8,4),(7,7)\}$.

Table 1. The reference points defined by experts within the course selection process.

A	Expert	G_1	G_2	F_1	F_2	F_3
$a_{1,1}$	I_1	2	1	1	0.8	0.9
$a_{1,2}$	I_2	1	3	1	0.9	0.7
$a_{2,1}$	I_1	7	3	0.8	0.6	0.6
$a_{2,2}$	I_2	5	6	0.7	0.7	0.7
$a_{2,3}$	I_2	4	7	0.6	0.7	0.9
$a_{2,4}$	I_2	6	7	0.6	0.5	0.7

After the expert recommendations are entered to the decision support system, their verification, fusion, and decision selection process runs as follows:

1. The *verification* of the conditions (2)-(4) discovers an internal inconsistency in the class A_2 while all other classes and their mutual situation are correct.
2. The *regularization* procedure starting from $a_{2,1}$ and $a_{2,4}$ replaces this pair with their average $a_{2,5}=(5;7)$. This new reference point turns out to be still comparable with $a_{2,2}$, so in the second step of this run both are replaced with $a_{2,6}=(5;6.5)$ that - together with $a_{2,1}$ yields a correct class A_2. The next run of averaging takes into account the second potential expert recommendation fusion, where $a_{2,2}$ and $a_{2,4}$ are compared first, so that a different average is calculated, namely $a_{2,8}=(5.5;6.5)$ that is noncomparable with $a_{2,1}$. Both runs yield correct classes A_2, so to calculate the utility function, the procedure averages all runs and retains the final class $A_2'=\{(6;2),(5.25;6.5)\}$. The trust coefficients of I_1 and I_2 are decreased proportionally to the distances between the final averaged point and the replaced values.

3. The *recommendation fusion* procedure extrapolates the utility ν of criteria G values from the level sets constructed as triagulations of reference points of each class; then the utility $\tilde{v}$, a function of decisions u, is expressed as $\tilde{v}=\nu\circ\varphi\circ F$.
4. The function φ is updated based on G and the new set A_2 wihin an unsupervised learning procedure and applied to find the course u_1 with the highest utility ν. The course u_1 such that $G(F(u_1))=(2,4)$ maximizes ν and is presented to the user.
5. The overall procedure stops when the proposed course is accepted, otherwise the above steps 1-4 are repeated with new expert recommendations and/or user-defined course parameters included into the classes A_i, for i=0,1,2,3. ■

Let us observe that all elements of reference classes are non-comparable with respect to F, although the regularization procedure was necessary when considering the criteria G. This explains the source of potential inconsistencies of expert judgments, who analyze the course features F, but present user-oriented assessments in terms of G.

The criteria of individual learners f may be different from G and not disclosed to instructors. They can serve as user preferences when selecting compromise courses.

4 Conclusions

The output of the above-presented decision making process may depend on the order the reference points and classes are taken into account. This is why the results can be averaged by performing the regularization for every permutation of points in each class. It can be shown that only permutation inside classes are relevant. If the number of recommending instructors is smaller than ten, and each of them defines no more than ten reference multimedia items, the above averaging is computationally feasible.

References

1. Górecki, H.; Skulimowski, A.M.J.: A Joint Consideration of Multiple Reference Points in Multicriteria Decision Making. *Found. Control Engrg.*, 11(2), 81-94 (1986).
2. Lai, Y.-J.; Liu T.-Y.; Hwang. C.-L.: TOPSIS and MODM. *EJOR*, 76, 486-500 (1995).
3. Rotter, P., Skulimowski, A.M.J.: Preference extraction in image retrieval. In: Ma, Z. (Ed.). *AI for Maximizing Content-based Image Retrieval*, IGI, Hershey, pp. 237-262 (2009).
4. Skulimowski, A.M.J.: Methods of Multicriteria Decision Support Based on Reference Sets. In: Caballero R., et al. (eds.) Advances in Multiple Objective and Goal Programming, *Lect. Notes in Econ. and Math. Syst.*, 455, Springer, pp. 282-290 (1997).
5. Skulimowski, A.M.J.: Cognitive content recommendation in digital knowledge repositories – a survey of recent trends. In: Rutkowski, L. et al. (eds.), 16th ICAISC, Proceedings, *Lect. Notes in Artif. Intell.*, 10246, 574–588 (2017), doi: 10.1007/978-3-319-59060-8_52.
6. Skulimowski, A.M.J., Rotter, P.: Applying reference sets in content-based interactive image retrieval. *Multiple Criteria Decision Making*, 4(SI, Trzaskalik, T., ed.), 185-202 (2009).
7. Trzaskalik, T.: Bipolar sorting and ranking of multistage alternatives. *Central Eur. J. Oper. Res.*, 29, 933–955 (2021), https://doi.org/10.1007/s10100-020-00733-2.
8. Vagliano, I., et al.: Open Innovation in the Big Data Era with the MOVING Platform, *IEEE MultiMedia*, 25(3), 8–21 (2018), https://doi.org/10.1109/MMUL.2018.2873495.

Machine Learning in Information Retrieval - Classification of Precision Medicine Documents

Jakub Dutkiewicz and Czesław Jedrzejek

Politechnika Poznańska, Plac Marii Skłodowskiej-Curie 5, 60-965 Poznań, Poland
jakub.dutkiewicz@put.poznan.pl
czeslaw.jedrzejek@put.poznan.pl

Abstract. This work is concerned with machine learning in Information Retrieval when features are not explicit and the relevance assessment process indicates that term frequency methods should be complemented with classification procedures. We apply our approach to the Text REtrieval Conference (TREC 2017) PM Track[1]. We concentrate on so-called Human PM criterion. The goal of this study is to present a method, which automatically obtains value of this feature. We present the effectiveness of a simple boolean search based method and a method of converting a semi-structured document into a vectorized form. The vectorized document form is then used by various machine learning methods in order to solve this task. We achieve an accuracy of 77.89% with the use of a Support Vector Classifier.

Keywords: Precision Medicine · Classification · Information Retrieval.

1 Introduction

Precision medicine is a medical model that separates people into different groups for which medical decisions, practices, and interventions are addressed. The amount of knowledge required by a physician to put the findings of precision medicine into practice is huge and the goal of information retrieval is to help the best possible treatment for a particular patient. The Precision Medicine (PM) Track of the Text REtrieval Conference (TREC) deals with providing detailed information necessary for PM care. There are two target document collections for the Precision Medicine track: scientific abstracts and clinical trials. Here, we are concerned only with clinical trials. The corpus is derived from ClinicalTrials.gov, a repository of past, present, and future clinical trials in the U.S. and abroad. A total of 241,006 clinical trial descriptions compose the corpus provided to participants.

The TREC PM 2017 was the first one in the series for which there were special rules for evaluation[2]. The result assessment starts with determination of the so-called *Human PM* feature: The clinical trial (1) relates to humans, (2) involves

[1] https://trec.nist.gov
[2] http://www.trec-cds.org/relevance_guidelines.pdf

some form of cancer, (3) focuses on treatment, prevention, or prognosis of cancer, and (4) relates in some way to at least one of the genes in the topic[1]. In this study, we focus on determining the value of that feature using Boolean search and machine learning methods. This is one of the key features in determining the final relevance of the document. It is used by annotators in the process of assigning the relevance of a document. This study focuses solely on determining this feature value, not on the impact of the feature on the final ranking.

2 Related Work

In the original TREC PM 2017 track, the best overall results for clinical trials were achieved by the UD_GU_BioTM and UTDHLTRI teams. UTDHLTRI [2] used the Precision Medicine Drug Graph (PMDG) system, for both Topic Analysis and Topic Expansion. The works of UTDHLTRI are similar to this work, as the authors of this work use external resources in order to find synonymic forms of the query terms and thus expand the queries. A similar Boolean search, which is tested in our study, was applied by [3] for TREC PM 2021. Works of [4] make use of the task of classifying documents as either relevant or irrelevant to clinical studies. However, they put focus on scientific abstracts, instead of clinical trials. Clinical trials and scientific abstracs are quite different systems.

3 Method Overview

The classical Information Retrieval process is defined as a regression task. Given a set of queries and a corpus of documents, the task of the IR process is to assign a relevance score for each document-query pair. A ranking list is built upon the score assignments. Documents with a higher score are ranked higher than documents with a lower score. Contrarily to the regression-based Information Retrieval, the classification-based Information Retrieval task is to assign the relevance value of a document to each query. Relevance values are defined as a finite and discrete set of literals, which correspond to classes in the classification process. The TREC Precision Medicine challenge, in our opinion, is more fit to the classification-based Information Retrieval task. This is due to the well-defined relevance assessment process, which is conducted by human annotators.

3.1 Document Structure

Each processed document describes a clinical trial and is structured as a shallow tree. A root of the tree indicates that the document is in fact a clinical trial, while the leaves relate to specific fields of the trial, such as the title of the trial, its description, eligibility criteria, outcome of the clinical trial, etc. We categorize fields it is comprised of as either informative or non-informative. We do not perform any calculations on the non-informative fields. The informative fields include “brief title”, “official title”, “keywords”, “condition”, “summary”,

"description", "intervention name", "eligibility criteria", "primary outcome", and "secondary outcome".

Another piece of information comes in a form of a topic. Each topic consists of four fields: "disease", "gene", "demographics" and "other". In the process of determining the Human PM feature of the document, only the "gene" field is relevant.

For each informative field, we create a dedicated processing function. The task of this function is to get to the informational portion of a field, tokenize the text value and return a list of tokens, of which the field is comprised. We preserve the information of the origin of the token by adding an adequate prefix to the token.

We first use processing functions to create a vector space for document vectors. Each unique token within the fields corresponds to a dimension within the space. We then use the processing functions in order to create a set of vectors for documents. If a document contains a word, which corresponds to a vector component, the value of that component is set to 1, it is set to 0 otherwise.

4 Conducted Experiments

We use an annotated sample of a Clinical Trials corpus. The entire sample consists of 13019 documents, out of which 3959 documents are annotated as *Human PM*. We use all of the positively annotated documents. We randomly pick 3959 documents annotated as *Not PM* in order to complement the dataset.

Table 1. Evaluation of expansion methods for the Boolean Search classification.

Method	Accuracy	Precision	Recall
No expansion	66.25%	46.37%	76.98%
Gene name synonyms	69.71%	63.52%	72.50%
Targetted drugs	68.50%	57.89%	73.49%
Drugs and genes	69.44%	68.00%	69.45%

We test the effectiveness of a Boolean Search for the classification. We create a set of dedicated queries, which consist of terms, which correspond four aspects of precision medicine. We expand the queries with use of the external tools, such as MeSH database [3]. We present the effectiveness of Boolean Search in Tab. 1.

We employ a set of classical machine learning methods in order to solve this task. In the implementation, we use the sklearn [5] library. We test the expansion methods described in the previous section. We use vectorized documents as an input and the PM class as an output of the classification procedure. Detailed information about the conducted experiment can be found in Tab. 2. It should be noted that used evaluation measures are dedicated for determining the quality

[3] https://www.ncbi.nlm.nih.gov/mesh/

Table 2. Accuracy of machine learning based classification methods.

Method	No expansion	Gene expansion	Gene and drug expansion
Boolean search	66.25%	69.71%	69.44%
k-NN	70.11%	70.43%	70.62%
Random Forests	75.31%	74.74%	75.34%
Decision Tree	71.11%	70.51%	70.95%
Gaussian Naive Bayes	68.48%	68.34%	68.23%
Multinomial Naive Bayes	76.24%	75.90%	75.76%
SVM	77.84 %	77.43%	77.34%

of a classification system and are not used to evaluate the quality of the final ranking.

5 Conclusions

The following study proves that machine learning methods do outperform query-based Boolean search methods in the task of determining the Precision Medicine aspect of a clinical trial. When designing a method of information retrieval and evaluating results, one is facing very many options of taking into account document tags, informative features, synonyms, and relations. One has to go through numerous combinations of these and the effects of these options on the overall results get blurred. Therefore, here we concentrate on one aggregate feature: Human PM. Surprisingly, relating genes to drugs did not improve results.

References

1. Roberts, K., Demner-Fushman, D., Voorhees, E.M., Hersh W.R., Bedrick S., Lazar, A.J., Pant, S.: Overview of the TREC 2017 Precision Medicine Track. In: Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA (2017)
2. Goodwin, T.R., Skinner, M.A., Harabagiu, S.M. : Will Sorafenib Help?: Treatment-aware Reranking in Precision Medicine Search. In: Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA (2017)
3. Rybinski, M., Karimi, S. : UTD HLTRI at TREC 2017: Precision Medicine Track In: CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia (2021)
4. Ševa J, Wiegandt DL, Götze J, Lamping M, Rieke D, Schäfer R, Jähnichen P, Kittner M, Pallarz S, Starlinger J, Keilholz U, Leser U. VIST - a Variant-Information Search Tool for precision oncology. BMC Bioinformatics (2019)
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research vol.12 2825–2830 (2011)

Rough Inclusion Based Toy Decision Systems Generator For Presenting Data Mining Algorithms*

Piotr Artiemjew[1][0000−0001−5508−9856]

Faculty of Mathematics and Computer Science, University of Warmia and Mazury in Olsztyn, 10-710 Olsztyn, Poland artem@matman.uwm.edu.pl
http://wmii.uwm.edu.pl/ artem/

Abstract. In this work, we address researchers who want to use various meaningful, real-world data to demonstrate their algorithms using toy examples. We propose a tool for generating toy decision systems from original decision systems addressing real decision problems. The toy decision systems are generated using a concept-dependent granulation technique and feature selection using the relative discernibility matrix. The resulting toy decision systems represent the essence of real decision systems, they provide meaningful data that can be used to demonstrate the action of data mining algorithms.

Keywords: Decision Systems · Toy Decision Systems · Generator · Data Mining · Tiny Data · Toy Data · Small Data · Rough Sets

1 Introduction

The article is a technical note presenting a novel toy decision system generator for use in presenting data mining algorithms. We present an idea for generating toy decision systems from real-life decision systems. The outcome of this work is a working generator for open use [1]. The tool has been developed using techniques derived from the rough set theory - [2]. In particular, we used the concept-dependent granulation technique to reduce the size of decision systems, a method derived from the technique proposed by Polkowski in [4] and developed in the paper [6]. To reduce the number of attributes, we used a relative discernibility matrix. The technique used is a method developed by Polkowski and further developed by Artiemjew for reducing the volume of decision systems while preserving their internal knowledge. A very extensive study in this context is carried out in the monograph [5].

1.1 Motivation

Probably every data scientist has come across Quinlan's toy decision system [7] (the toy decision problem of playing tennis), which has become a popular deci-

* This work has been supported by the grant from Ministry of Science and Higher Education of the Republic of Poland under the project number 23.610.007-000

sion system for presenting toy examples of ideas for new data science algorithms. Quinlan used this decision system to demonstrate the ID3 technique, the foundation for the C4.5 method [7]. The idea to build an application for generating small meaningful decision systems comes from the fact that there is a lack of such a tool on the Internet. The author has built it for his own use and wants to share its functionalities with other researchers.

The rest of this work consists of the following sections. In section 2 we have a detailed description of the methodology used. In section 3 we have a demonstration of the generator. In section 4 we have a summary of publications and future plans.

2 Methodology

A general scheme for creating small decision systems is to granulate the original data and to reduce dimensions. We will first introduce the method of reducing the number of objects of decision systems.

2.1 Granulation Techniques - Reduction of the Number of Objects

Our methods are based on rough inclusions. Introduction to rough inclusions in the framework of rough mereology is available in Polkowski [3] – [5]. We start with a detailed description of the basic method - see [4] - standard granulation. Let us consider the decision system (U, A, d), where U is the universe of objects, A the set of condition attributes, $d \notin A$ is the decision attribute, and r_{gran} granulation radius from the set $\{0, \frac{1}{|A|}, \frac{2}{|A|}, ..., 1\}$. The standard rough inclusion relation μ, for $u, v \in U$ and for selected r_{gran} is defined as

$$\mu(v, u, r_{gran}) \Leftrightarrow \frac{|IND(u,v)|}{|A|} \geq r_{gran}, \text{ where } IND(u,v) = \{a \in A : a(u) = a(v)\}, \tag{1}$$

For each object $u \in U$, and selected r_{gran}, we compute the *standard granule* $g_{r_{gran}}(u)$ as follows, $g_{r_{gran}}(u)$ is $\{v \in U : \mu(v, u, r_{gran})\}$. In the next step, we use selected strategy to cover the training decision system U by computed granules - the random choice is the simplest among the most effective studied in [5]). All methods being studied are available in [5] (pages 105 – 220). And in the last step, granular reflection of training set is computed with the use of Majority Voting procedure. The ties are resolved randomly. A concept–dependent (cd) granule $g^{cd}_{r_{gran}}(u)$ of the radius r_{gran} of u is defined as follows:

$$v \in g^{cd}_{r_{gran}}(u) \text{ if and only if } \mu(v, u, r_{gran}) \text{ and } (d(u) = d(v)) \tag{2}$$

2.2 Attribute Selection Using a Relative Discernibility Matrix

We used the following tool to select the attributes of the original decision system. For the decision system (U, A, d), where $A = \{a_1, a_2, ..., a_k\}$, $a_i \in A$ we define matrix grids, for pairs of objects $u_{j_1}, u_{j_2} \in U$.

$$c_{j_1 j_2}^{a_i, relative} = \begin{cases} 1, \ if \ a_i(u_{j_1}) ! = a_i(u_{j_2}), d(u_{j_1}) ! = d(u_{j_2}), j_1 < j_2 \\ 0, \ otherwise \end{cases} \tag{3}$$

$class\_separation\_level_{a_i} = \sum_{u_{j_1}, u_{j_2} \in U} c_{j_1 j_2}^{a_i, relative}$

After calculating the *class_separation_level* for all condition attributes, we rank them and select a fixed number for the final decision system. The higher the value of the *class_separation_level* parameter, the better the class separation.

3 Demostration of the Generator

To demonstrate the operation of our tool, we selected several decision systems from the UCI repository [8]. We implemented our tool in django technology. The detailed appearance of our application available at [1], can be seen in Figure 1. The application is divided into three parts - input, intermediate and output layer. The left window shows information about sample data and recommended parameters. The second window is for entering parameters. In the field *Link to your decision system* we have space to paste a link to the data, it can be in csv (comma-separated values) or tsv (tab-separated values) format. In the *Class size of the data drawn* field we can specify the number of objects of particular classes to be drawn from the original decision system. This step allows us to process large decision systems. The *No. of attributes to extract*? field specifies the number of attributes to be extracted. In the field *Degree of approximation* is the number of the degree of approximation, which is from the set $\{0, 1, \ldots, |A|\}$. The last window is used to display the resulting decision system.

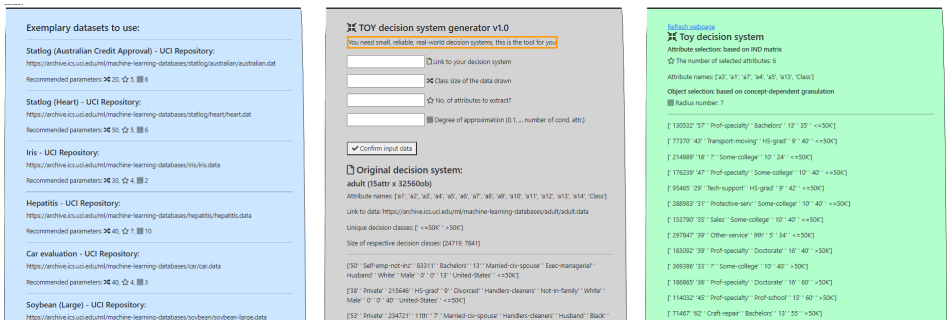


Fig. 1. Application screenshot. The middle window is used for data entry, the left window is information about the decision systems that have been prepared to test the application, the right window is the resulting window with the extracted toy decision system.

In Tab. 1, one can see an example of the effect of our generator - toy decision systems generated based on the Iris and Car ([8]) data sets.

Table 1. An example of toy decision systems generated with our application.

From the Iris data set

a3	a4	a1	Class
1.5	0.1	5.2	Iris-setosa
4.2	1.5	5.9	Iris-versicolor
5.8	2.2	6.5	Iris-virginica

From the Car data set.

a3	a2	a1	a5	Class
4	med	vhigh	big	acc
5more	high	low	big	acc
4	med	high	small	acc
2	low	med	big	good
3	med	low	small	good
4	low	med	med	unacc
4	vhigh	vhigh	big	unacc
5more	high	low	med	unacc
2	high	low	big	vgood
4	low	low	big	vgood
4	low	med	med	vgood

4 Conclusions

The current paper presents a tool for creating toy decision systems from large real-world data. To achieve this goal, a decision system approximation technique - the concept-dependent granulation method - and a feature selection method based on a relative discernibility matrix are used. This tool is dedicated to researchers who want to present their data science algorithms on small meaningful data. The version we present is dedicated to symbolic data. The tool works on sample recommended parameters. In the future, we plan to extend the tool to generate decision systems for data of any type. Additionally, a desktop version using tkinter technology is being developed.

References

1. Toy decision system generator, http://toyds.herokuapp.com/generator/v1/. Last accessed 12 Apr 2022
2. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341—356 (1982). https://doi.org/10.1007/BF01001956
3. Polkowski, L.: Formal granular calculi based on rough inclusions, In: Proceedings of IEEE 2005 Conference on Granular Computing GrC05, pp. 57–62. IEEE Press, Beijing, China (2005)
4. Polkowski, L.: A model of granular computing with applications, In: Proceedings of IEEE 2006 Conference on Granular Computing GrC06, pp. 9–16. IEEE Press, Atlanta, USA (2006)
5. Polkowski, L., Artiemjew, P.: Granular Computing in Decision Approximation - An Application of Rough Mereology, In: Intelligent Systems Reference Library 77, Springer, ISBN 978-3-319-12879-5, pp. 1–422 (2015).
6. Artiemjew, P.: Classifiers from Granulated Data Sets: Concept Dependent and Layered Granulation, In: Proceedings RSKD'07. The Workshops at ECML/PKDD'07, pp. 1–9., Warsaw Univ. Press, Warsaw (2007)
7. Quinlan, J., R.: C4.5: Programs for Machine Learning, Morgan Kaufmann, Kluwer Academic Publishers (1993). https://doi.org/10.1023/A:1022645310020
8. UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php. Last accessed 12 Apr 2022

Selected Applications of Graph-based Knowledge Representation

Weronika T. Adrian[0000−0002−1860−6989], Krzysztof Kluza[0000−0003−1876−9603], Mateusz Zaremba[0000−0002−3888−4783], Paweł Jemioło[0000−0001−5962−4043], Marek Adrian[0000−0002−0435−0994], Antonia Pękala, Kinga Florek, and Antoni Ligęza[0000−0002−6573−4246]

AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
{wta,kluza,mzaremba,pawljmlo,madrian,ligeza}@agh.edu.pl

Abstract. Graph-based conceptualization constitutes a natural mean for knowledge representation in several branches of Artificial Intelligence. From ontological descriptions of the universe of discourse to graphical depiction of business processes, graphs play an important role in modern intelligent systems. In this submission, we report on some of the research threads on graph-driven knowledge engineering carried out within KRaKEn Research Group at AGH University of Science and Technology.

Keywords: Knowledge Representation · Knowledge Engineering · Explainability · Transparency · Graphs

1 Introduction

Graphs, in their simple form a pair of sets of nodes and edges, are a mathematical abstraction that has found numerous applications in Computer Science as a convenient data structure and flexible representation method. In this paper, we review some of the recent developments in selected branches of knowledge management and engineering using graphs in KRaKEn (https://kraken.edu.pl/).

The paper is structured as follows. In Section 2, we present a graph-based solution for analysis of the relations in our research group. Section 3 shortly introduces our approach to using graphs for AI explainability. Section 4 provides an insight to our future works regarding using graph techniques for analyzing business process models. We conclude our paper in Section 5.

2 Graphs for Knowledge Representation and Processing

In order to explore the possibilities of knowledge representation and visualization with graphs, a use case of a research group has been analyzed [1]. A network of relations among team members, activities, research areas and teaching responsibilities has been modeled with a graph database (see Fig. 1), thus enabling flexible visualization possibilities (see Fig. 2) based on simple queries.

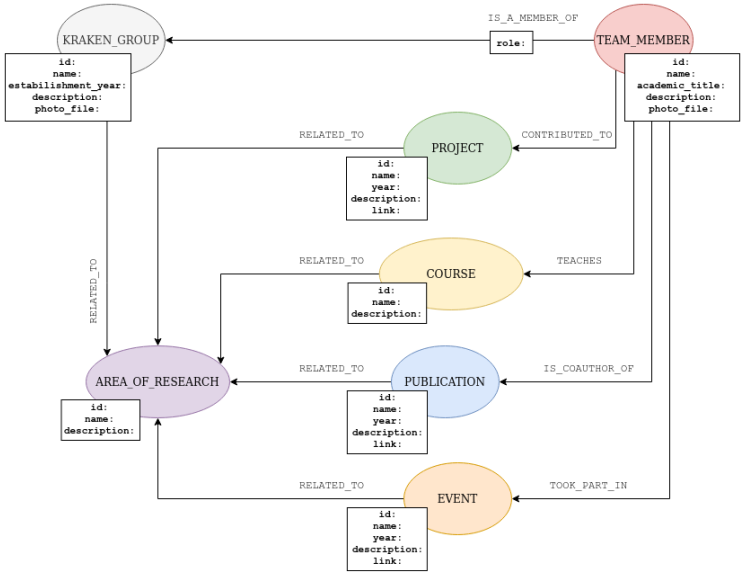


Fig. 1. Graph-based knowledge representation about the research team.

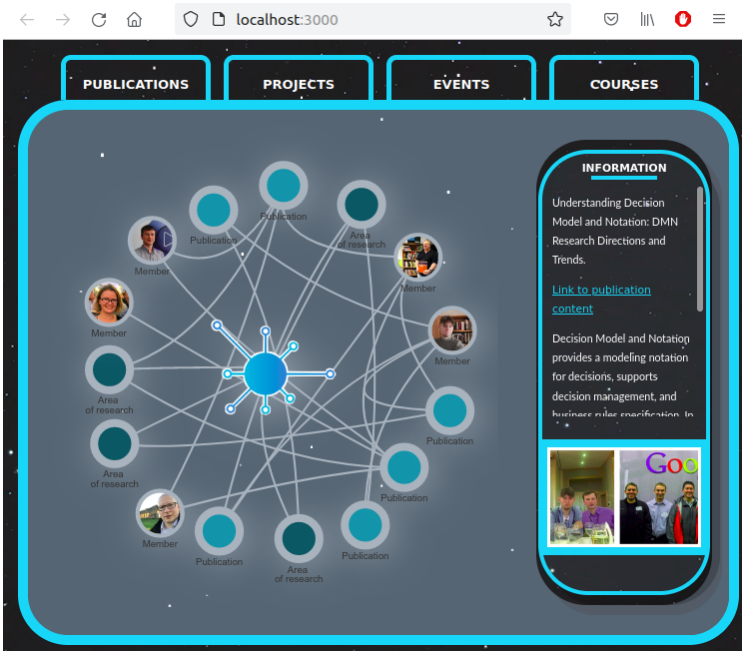


Fig. 2. View of a publication with links to its authors, metadata etc.

Possibilities of using graph-oriented knowledge has also been explored within the context of recommendation systems [2]. Both knowledge-based recommenders as well as content-based filtering systems can be conveniently designed and implemented when knowledge is modelled with graphs. In fact, recommendations can extensively use simple queries enriched with some similarity calculations.

```
MATCH (r:Resort)
WITH COLLECT({resort: r, score: 0}) AS resorts

MATCH (r:Resort)-[a:ACCOMMODATION]-(:Accommodation)
WHERE a.halfBoard <= 1000 OR a.fullBoard <= 1000
WITH DISTINCT r AS r, resorts, 4.0 AS s
WITH resorts + COLLECT({resort: r, score: s}) AS resorts

MATCH (r:Resort)-[:WINE_AND_DINE]-(w:WineAndDine)
WHERE w.type IN ["Underwater Restaurant", "Bar"]
WITH r, resorts, COUNT(w) * 5.0 / 2 AS s
WITH resorts + COLLECT({resort: r, score: s}) AS resorts

UNWIND resorts AS record
RETURN record.resort AS r, sum(record.score) / (4.0+5.0) * 100
AS score
ORDER BY score DESC
```

3 Graphs for Explainability

Despite providing high performance, deep learning techniques are often perceived as complicated and untrustworthy. To mitigate this issue, the domain of explainability and interpretability arouse. Its goal is to generate outputs along with explanations, which are understandable for experts and lay people.

Witch several methods existing, none of them utilised graphs. In [3], we propose a novel approach for explainable systems, which incorporates knowledge graphs for presentation of results. A node in such a graph stays for influence of a given feature and the thickness of its edge depicts importance of a specific connection. The use of graphs allows for highlighting two variables and their interactions.

4 Graphs for Business Process Analysis

Models of business processes provide knowledge representation that helps organizations visualize and optimize their processes, allowing them to achieve their business goals more effectively. Such models can be represented using various notations [4, 5]. Process model discovery is a group of popular methods for knowledge acquisition under the broad umbrella of process mining solutions [6]. Such mining techniques can generate models from the event logs without any *a priori* knowledge about the process.

In recent years, modern versions of the algorithms which allow for BPMN models discovery has been proposed, such as BPMN Miner [7], Split Miner [8], P-Miner [9]. As such discovered models constitute graphs, it is possible to compare and analyze the obtained models using graph similarity methods [10] and take advantage of the knowledge graphs techniques for such analysis [11]. Moreover, the classical process discovery algorithms work on the event logs. Graph based algorithms might be also helpful in enhancing the data [12], especially unstructured data for creating event logs.

5 Conclusions

In this paper, we give a brief overview of the selected research threads conducted in the KRaKEn Research Group at the AGH UST concerning graph-based knowledge representation. We show how graphs can be used for knowledge representation and processing, provide the explainability method, as well as be utilized for business process analysis.

References

1. Pękala, A.: The research team's website designed based on the knowledge graph [Bachelor thesis] (2022)
2. Florek, K.: A knowledge graph-based recommendation system for hotels [Bachelor thesis] (2022)
3. Mróz, P., Quemy, A., Ślażyński, M., Kluza, K., Jemioło, P.: Gbex-towards graph-based explanations. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). pp. 112–117. IEEE (2020)
4. Kluza, K., Wiśniewski, P., Jobczyk, K., Ligęza, A., Mroczek, A.S.: Comparison of selected modeling notations for process, decision and system modeling. In: 2017 Federated Conference on Computer Science and Information Systems (FedCSIS). pp. 1095–1098. IEEE (2017)
5. Suchenia, A., Łopata, P., Wiśniewski, P., Stachura-Terlecka, B.: Towards UML representation for BPMN and DMN models. In: MATEC Web of Conferences. vol. 252, p. 02007. EDP Sciences (2019)
6. van der Aalst, W.: Process Mining: Data Science in Action. Springer Berlin Heidelberg (2016)
7. Conforti, R., Dumas, M., García-Bañuelos, L., La Rosa, M.: BPMN miner: Automated discovery of BPMN process models with hierarchical structure. Information Systems **56**, 284–303 (2016)
8. Augusto, A., Conforti, R., Dumas, M., La Rosa, M.: Split miner: Discovering accurate and simple business process models from event logs. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 1–10. IEEE (2017)
9. Marin-Castro, H.M., Tello-Leal, E.: An end-to-end approach and tool for BPMN process discovery. Expert Systems with Applications **174**, 114662 (2021)
10. Adrian, W.T., Skoczen, S., Majkut, S., Kluza, K., Ligeza, A.: Tracing the evolution of approaches to semantic similarity analysis. In: KEOD. pp. 157–164 (2020)
11. Adrian, W.T., Adrian, M., Kluza, K., Stachura-Terlecka, B., Ligeza, A.: Extended knowledge graphs: A conceptual study. In: KEOD. pp. 173–180 (2020)
12. Bhatt, S., Sheth, A., Shalin, V., Zhao, J.: Knowledge graph semantic enhancement of input data for improving ai. IEEE Internet Computing **24**(2), 66–72 (2020)

Styles with Benefits. The StyloMetrix Vectors for Stylistic and Semantic Text Classification of Small-Scale Datasets and Different Sample Length

Inez Okulska[0000-0002-1452-9840] and Anna Zawadzka[1]
[1] NASK National Research Institute, Warsaw, Poland
inez.okulska@nask.pl

Abstract. This paper offers a new approach to the problem of text classification of small-sized datasets and datasets with text samples of different lengths. StyloMetrix is a tool that allows for representing a text sample of any length with a linguistic vector of a fixed size – with 138 features. Each feature in the vector translates into a quantified, reproducible measure of an objective linguistic phenomenon – i.e., normalized statistics of chosen occurrence of morpho-syntactic or lexical relations. Thus, StyloMetrix vectors also show great potential in explainable AI. The StyloMetrix vector can serve as input data for classification algorithms, achieving high accuracy for relatively small-scale datasets with samples of varying lengths, compared to the semantic embeddings. Even though StyloMetrix does not encode any semantic information, the vectors (despite its name) also proved to be a valuable tool for content classification.

Keywords: Text Classification, Stylometry, Document Embeddings.

1 Vector Representations for Text Classification

1.1 Introduction

One of the challenges of text classification is the right choice of text representation, trading off the performance, resources, adjustments to the dataset or explainability. The most popular Word2Vec [8], Glove [10], or FastText [1] embeddings still amaze us with their universality for various tasks. However, the semantic relations between words depend strongly on the training particular corpus, and homonyms are represented by only one vector (no disambiguation). The BERT [2] embeddings are contextual and much more precise. However, even the pre-trained models often require extensive training data to be fine-tuned for a new task. Moreover, word embeddings present the problem of sample size, which needs to be equal or padded or trimmed, which is challenging for datasets with a wide range of sample lengths.

This paper proposes an extended, proprietary, stylometric vector representation obtained with the StyloMetrix tool: the StyloMetrix vectors. Besides resolving the issue of varying sample length (the vectors encode entire documents, no matter the size), they offer a different perspective on the represented text, i.e., a linguistic one. Not the "meanings" of the words are encoded, but the stylistic structure of the entire sample. StyloMetrix vectors have proved to have good performance not only for style, but

also for content classification – the vectors are sensitive to semantics, even for small-scale datasets with samples of varying lengths.

1.2 Related Work

Stylometric research typically tackles authorship attribution and semantic classification tasks. The focus is often on words frequencies or stopwords incidences [13][14]. For the Polish language, such a lexical approach is taken in the stylo package for R created by Eder and Rybicki [3], which offers text representation as vectors of relative frequencies of extracted n-grams. For reliable performance, texts need to be of 5000 tokens at minimum. Representing entire documents with fixed-size vectors has been proposed in Doc2Vec [7], i.e., an enhanced CBOW-like method based on Word2vec (semantic) word embeddings, and by Ryciak et al. l [12], who introduced word histograms that produce normalized vectors of 9000 features. It has shown great potential for unsupervised classification tasks of an extensive, unbalanced data set of patent documents. As for the morphosyntactic approach, the Websty tool [11] for the Polish language offers document comparison by manual feature selection from computed distributional metrics. However, it does not cover syntactic structures.

2 The Method (The StyloMetrix Vectors)

StyloMetrix is a tool that allows for representing a text sample of any length with a linguistic vector of a fixed size – 138 features. Each feature in the vector translates into a quantified, reproducible measure of the chosen occurrence of morpho-syntactic or lexical relations. The metrics so far include: a) POS classification, b) grammatical forms of nouns, pronouns, c) comparative degrees in adjectives and adverbs, d) verbs in grammatical persons, aspect, tense, mood, in participle form, e) type-token ratio, f) incidence of content and function words, g) incidence of names and personal names, h) volume of 1%/5% of most common types for text, i) psycholinguistic metrics related to an experimental study by Imbir [4], j) incidence of declarative, interrogative and exclamative sentences, k) participation of words in nominal phrases and modifiers, l) sentence starts, syllable length and m) the presence of categorical affixes. The metrics are normalized, i.e. their values are relative to the total number of words (alphanumerical tokens) in a sample, always providing a value in the range [0, 1]. They relay on morphosyntactic information in the spaCy model for the Polish language [15].

The StyloMetrix vector can serve as input data for classification algorithms, achieving high accuracy for relatively small-scale datasets with text samples of unbalanced length. The encoding does not transmit any semantic information as it is usually not subject to interest in stylometric experiments. However, it also proved to be a valuable tool for content classification. The example of 10 metrics computed for four excerpts from the dataset used further in the first experiment are shown in Table 1. To validate the metric values in the Table 1, we present two from the analyzed excerpts, one from non-professional erotic stories (A) and non-professional neutral short stories (B), accordingly. The trends presented on these random samples mostly represent their classes, hence we decided to use them for underlining the possible contrast.

It seems that a certain combination of linguistic choices (e.g. the range of vocabulary, verbs over nouns, the use of contemporary transgressives, or 1st person singular) can denote semantic aspects specific to a text genre, i.e. erotic stories in this case.

Table 1. Chosen metrics for random excerpts from each category

Class	Sample 1 n-prof erotic	Sample 2 n-prof neutral	Sample 3 non-fiction	Sample 4 pop lit
Verbs	**0,27**	0,19	0,18	0,23
Nouns	0,27	0,32	0,42	0,37
Pronouns	**0,20**	0,12	0,03	0,07
1st person singular verbs	**0,17**	0,09	0	0
3d person singular verbs	0,03	0,05	0,13	0,20
Verbs in past tense	**0,17**	0,12	0,01	**0,20**
Contemporary transgressives	**0,07**	0	0	0
3rd person singular pers. pronouns	**0,10**	0	0,01	0,03
Tokens covering 1% of most common types	**0,17**	0,09	0,11	0,10
Words in a nominal phrase	0,30	0,63	0,73	0,67

A: *Uśmiechnęłam się do niego. Ponownie przeciągnęłam językiem po czubku penisa, obserwując, jak Adam przymyka powieki. Rozchyliłam wargi i naprowadzając dłonią penis, powiodłam nim po nich. Wreszcie wsunęłam go do ust.*

B: *W międzyczasie zjadłem obiad, później założyłem elegancką koszulę. Z biegiem czasu jednak robiło się niepokojąco cicho. Postanowiłem dowiedzieć się od współlokatorki Alicji, jak z przygotowaniem na imprezę i o której godzinie ma zjawić się Magda. Jednak nie uzyskałem odpowiedzi od razu!*

3 The Experiments

Two classification tasks were tackled: text genres with or without adult content and sentiment analysis, both on small-scale datasets. We used StyloMetrix vectors as input to Random Forests. Text genre classification has also been compared with the results of a RNN with BiLSTM layer and the BERT model.

We gathered four classes of samples of varying lengths (short stories, novels, reportages), including non-professional erotic stories and non-professional neutral stories, non-fiction, and pop literature (for statistics see Table 2). The data was split 80:20 for training and test. Each text sample was represented with the StyloMetrix vector of size 138. For the LSTM and BERT models, we split the data into text chunks of 250 tokens, with punctuation. Then we represented the chunks accordingly with vectors of size 100 (Polish Word2vec)[5] and 1024 (HerBERT large case)[9].

Table 2. Dataset statistics for the supervised text classification experiment

Class	n-prof erotic	n-prof neutral	non-fiction	pop lit	sum
Sample count	263	300	**556**	189	1308
Min sample length	215	304	**42**	543	-
Max sample length	5551	9825	40363	**130220**	-
Chunks count	801	1998	1108	**5234**	**9141**

The first experiment, i.e. supervised text classification with Random Forests and StyloMetrix vectors representing the text genres dataset (1308 samples) yielded the mean accuracy of 93.5%, offering balanced scores across classes as shown in Table 3. Classification using deep learning models on the same data resulted in poor performance, however the adjusted dataset of chunks (creating a bigger data set of 9141 samples) yielded 83.5% accuracy for LSTM and 84.25% for the HerBERT model.

Table 3. Supervised four text genres classification results for different word/text representation

Accuracy per class	n-prof erotic	n-prof neutral	non-fiction	pop-lit
RF with StyloMetrix (original samples)	**0.91**	**0.92**	**1.0**	0.91
BiLSTM with Word2vec (70 ep., 250-token chunks)	0.85	0.79	0.81	0.93
HerBERT large (200 epochs, 250-token chunks)	**0.91**	0.67	0.74	**0.95**

For the second experiment we used a subset of 607 samples of opinions in 2 domains (school and medicine) from the PolEmo 2.0 Sentiment Analysis Dataset [6]. We preprocessed the data to correct frequent typing errors in verbs (eg. *popatrzył em*) and used StyloMetrix to produce a separate NumPy matrix file (.npy) for each group[1].

Table 4 presents the results of the second experiment: classification of samples belonging to 4, 3, and 2 sentiment groups on the small subset and the full set. The ambivalent class turned out to be indistinguishable from positive samples in style in the current experimental setup. That is why we decreased the number of the classes – first excluding only the ambivalent category, and then the neutral as well.

The results indicate the best classification level for binary classification using the StyloMetrix data with a very simple classifier. It shows that sentiment can be traced in stylistic features, not only the semantic ones (like in most of the related approaches).

Table 4. Supervised sentiment classification with StyloMetrix vectors

Accuracy per class	Positive	Ambivalent	Neutral	Negative	Mean acc
4 classes – subset	0.81	0	0.54	0.58	0.48
3 classes – subset	0.88	-	0.62	0.76	0.75
2 classes – subset	**0.89**	-	-	**0.86**	0.86

4 Conclusions

The presented StyloMetrix vectors performed well on small-scale datasets (about 200-300 samples per class) with texts of different lengths, obtaining good results in classifying text genres (including separating the adult content from neutral ones) and sentiment in clients reviews. The results confirm our initial claims: StyloMetrix vectors provide novel, rich and qualitative representation of textual data that can even be

[1] The StyloMetrix vectors for this particular dataset may be downloaded from https://github.com/ZILiAT-NASK/Datasets/tree/main/StyloMetrix_PolEmo.

sensitive to semantics. We are thrilled to find out about further usages of this data as we extend the scope of metrics and try different experimental scenarios.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the association for computational linguistics 5, 135-146 (2017).
2. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).
3. Eder, M., Rybicki, J., Kestemont, M.: Stylometry with R: a package for computational text analysis. The R Journal 8(1), 107-121 (2016).
4. Imbir, K. K.: Affective norms for 4900 Polish words reload (ANPW_R): Assessments for valence, arousal, dominance, origin, significance, concreteness, imageability and, age of acquisition. Frontiers in psychology 7, Article 1081 (2016).
5. Kędzia, P., Czachor, G., Piasecki, M., Kocoń, J.: Vector representations of polish words (Word2Vec method). CLARIN-PL digital repository, http://hdl.handle.net/11321/327, last accessed 14/04/2022. (2016).
6. Kocoń, J., Miłkowski, P., Zaśko-Zielińska, M.: Multi-level sentiment analysis of PolEmo 2.0: Extended corpus of multi-domain consumer reviews. In: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pp. 980-991. Association for Computational Linguistics, Hong Kong (2019).
7. Lau, J. H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv:1607.05368 (2016).
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
9. Mroczkowski, R., Rybak, P., Wróblewska, A., Gawlik, I.: HerBERT: Efficiently pretrained transformer-based language model for Polish. arXiv preprint arXiv:2105.01735 (2021).
10. Pennington, J., Socher, R., Manning, C. D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. Associatio for Computational Linguistics, Doha (2014).
11. Piasecki, M., Walkowiak, T., Eder, M.: Open stylometric system WebSty: integrated language processing, analysis and visualization. Computational Methods in Science and Technology 24(1), 43-58 (2018).
12. Ryciak, N., Chrabąszcz, M., Bartoszuk, M., Classification of patent applications, an oral seminar held at the Institute of Computer Science, Polish Academy of Sciences (IPI PAN) available online as recorded video (www.youtube.com/watch?v=L8RRx9KVhJs) (2020)
13. Rybicki, J., Heydel, M.: The stylistics and stylometry of collaborative translation: Woolf's Night and Day in Polish. Literary and Linguistic Computing 28(4), 708-717 (2013).
14. Savoy, J.: Machine Learning Methods for Stylometry. Springer, Cham (2020).
15. Tuora, R., Kobylinski, Ł.: Integrating Polish language tools and resources in Spacy. In: Proceedings of PP-RAI 2019 Conference, pp. 210-214. Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology, Wrocław (2019).
16. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers), pp. 207-212. Association for Computational Linguistics, Berlin (2016)

A Protocol for Acquisition of Emotion and Affect in Music – Dataset Collection During COVID-19 (EAM-DCDC)

Aleksandra Wołowiec and Paweł Jemioło
AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
{wolowiec, pawljmlo}@agh.edu.pl

Abstract. Emotion and affect recognition constitute one of the most frequently explored areas in Human-Computer Interaction. Since the credibility of data used for generating Artificial Intelligence models in the field is questioned, acquiring new high-quality datasets is essential. In this study, we aim to present a protocol for the experiment, which focuses on gathering cardiovascular data during music stimulation. Following this protocol, our objective is to provide a credible, ecologically valid and publicly available dataset, which will be further used for emotion and affect recognition with Artificial Intelligence models. Preliminary insights from our pilot research serve as the basis for protocol.

Keywords: Music · Cardiovascular · Heart Rate · Emotion and Affect Recognition · Database · Dataset · COVID-19

1 Introduction

Emotion and affect recognition from physiological signals are one of the main objectives of Human-Computer Interaction. For this purpose, the internal states of participants are evoked employing different stimuli, e.g. an audio file with sounds of nature, music, pictures, or a short movie. At the same time, people that take part in experiments are connected to devices, which measure the activity of their brain, heart, or vascular system, to name only a few. Further, data are pre-processed, and they serve as an input for Artificial Intelligence models for emotion and affect recognition.

As our recent systematic review [5] revealed, there are 18 publicly available datasets for emotion and affect recognition, which contain cardiovascular signals. However, only 2 of them may be considered high-quality ones. In VRAD [6], authors used virtual reality for emotion elicitation, while AMIGOS [3] employed audiovisual clips. Therefore, there is still a need for new credible datasets, including cardiovascular signals, to be acquired with other stimulation methods, e.g., music or pictures.

The replication crisis in psychology and reproducibility problems in computer and information sciences show that the data used to generate Artificial Intelligence models need to be collected via thoroughly planned experiments [5]. Otherwise, we may face the *garbage in, garbage out* problem, which refers to a situation when doubtful input results in producing questionable output. That is why many initiatives in this field emerged, e.g. preparation of a protocol for the study and its registration, *registered reports*, preregistration of hypotheses, or replication of previously published studies.

Additionally, Human-Computer Interaction faces another issue, which arose from COVID-19. Namely, conducting research with human subjects in laboratories during a pandemic is arduous or even impossible due to restrictions. Fortunately, alternatives exist, and it occurs that most of the experiments may be successfully carried out via the internet thanks to initiatives like Amazon Mechanical Turk or simply through passing equipment to participants, particularly when the research does not involve highly-specialised apparatus. Not only the safety of the subjects is ensured, and they may participate in the experiment at a time that suits them, but also the ecological validity of the study increases.

The purpose of this paper is to present a protocol for the experiment, which focuses on the acquisition of physiological data during music stimulation. Following this protocol, we aim at providing a credible, ecologically valid and publicly available dataset, which will be further used for emotion and affect recognition with Artificial Intelligence models.

2 Methods

In this study, participants will experience music stimuli (songs excerpts). However, the choice of emotion-evoking material needs to be justified in a separate pilot study with other yet similar participants. One needs to make sure that selected stimuli indeed evoke intended emotions.

In our first attempt at stimuli validation, we selected 60 songs from the dataset with music excerpts grouped by the Artificial Intelligence model[1] into 4 emotional quadrants (High Arousal High Valence – HAHV, High Arousal Low Valence – HALV, Low Arousal Low Valence – LALV, Low Arousal High Valence – LAHV) of Russell space. Russell's space consists of two emotional dimensions, namely arousal and valence [7]. The first one refers to the intensity of emotion, while the second one is connected with the aspect of pleasure. Unfortunately, our initial choice occurred to be improper (see section 3); thus, we incorporated another source of stimuli for validation. For this phase we selected 64 songs from DEAM [1], 16 excerpts per Russell's quadrant. From 64 songs, we will choose 32 stimuli with the most dominant valence and arousal values, 8 per category.

In our experiments, the affective rating system called Self-Assessment Manikin (SAM) [2] will be used for stimuli assessment for all participants. SAM allows measuring three emotional dimensions of valence, arousal, and dominance with different scales. Each of the scales consists of 5 drawings showing the experienced emotional states. A verbal description will precede the scales for precise interpretation of the images. The user's task will be to select an adequate value from 1-9 using the slider.

We will include participants aged 20-30 with higher education (or ongoing) who have not graduated from music schools or work in the music industry but listen to music every day. Exclude criteria involve: cardiovascular or mental disorders and inability to recognise emotions (alexithymia). We plan to examine 45 people in the main study, which is more than average in existing datasets. See our recent systematic review for more details regarding the number of subjects [5].

[1] github.com/cristobalvch/Spotify-Machine-Learning (Accessed on 28.02.2022).

Additionally, we will control potential confounders. Firstly, we will ask participants about their mood, which may influence experienced emotions. Next, as the experiment will take place at the subjects' homes, they will be prompted to certify that they were listening to the songs themselves, no one disturbed them, and they were not taking any additional activities. Moreover, we will ask participants to rate their concentration level during the whole procedure.

The pilot and main studies will be conducted using Google Forms, and PsychoPy deployed on Pavlovia servers. Participants will receive instructions in Google Docs with links to both parts of the experiment and information on how to handle the delivered device (in case of the main study).

The first part is a personal questionnaire (Google Forms) with basic information: age, gender, ethnicity, which genre of music the respondents prefer, whether they usually listen to music in line with their mood or the opposite and how they feel at a given moment (SAM).

The second part consists of the following sequences: 10-second break, a 45-second song, SAM scale rating, and yes/no question for determining whether participants liked the song. The order in which the songs are played for each subject is determined randomly.

In the main experiment, the participant's heart rate will be measured with a wearable smart-band Xiaomi Mi Band 5 during the second part (PsychoPy). Before starting the experiment, subjects will be asked to prepare a phone with a voice recorder (recording experiment for synchronisation purposes) and a computer with an active loudspeaker. After starting the experiment, participants will be asked to wait 2 minutes for the band to measure their baseline. After the subject completes the experiment, the data will be extracted from the band using the Gadgetbridge application[2] for Android smartphones, and then the band will be passed on to the next participant.

After acquiring the data, we will validate the data (to confirm they can be used for emotion and affect recognition) using basic Artificial Intelligence algorithms. We will use Support Vector Machines, Naive Bayes, Random Forest and Deep Learning techniques (e.g. following Harper and Southern [4] method) as they are most frequently incorporated [5].

After validation, the data will be shared anonymously in the form of CSV files via Open Science Framework (OSF). Every participant will be asked whether they agree to share the data.

3 Preliminary results

In Figure 1, we present results regarding our first attempt to pilot study (valence and arousal ratings). Songs selected precisely for particular quadrants do not overlap with participants' answers. Notably, we have not obtained space coverage for LVHA songs that comply with the dataset ground-truth. We thus need another pilot study with songs chosen more adequately (the DEAM dataset [1]).

[2] codeberg.org/Freeyourgadget/Gadgetbridge (Accessed on 28.02.2022).

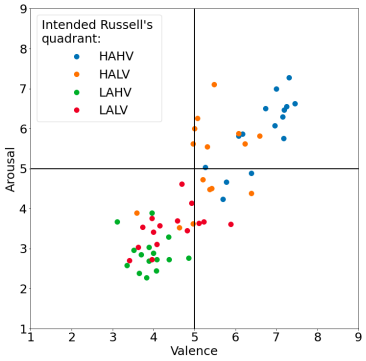


Fig. 1. Participants responses regarding heard songs (each song is a dot).

4 Conclusions and Future Works

In this paper, we comprehensively presented a protocol for the experiment, which focuses on the acquisition of cardiovascular data during music stimulation for emotion and affect recognition with Artificial Intelligence models.

Now, we want to follow this protocol, gather the data, validate it with selected models and share it. We believe that only thorough methodology allows for acquiring high-quality data for development of Artificial Intelligence solutions.

Funding Sources

Research project supported by program *Excellence initiative -- research university* for the University of Science and Technology.

References

1. Aljanaki, A., Yang, Y.H., Soleymani, M.: Developing a benchmark for emotional analysis of music. PloS one **12**(3), e0173392 (2017)
2. Bradley, M.M., Lang, P.J.: Measuring emotion: the self-assessment manikin and the semantic differential. Journal of behavior therapy and experimental psychiatry **25**(1), 49–59 (1994)
3. Correa, J.A.M., Abadi, M.K., Sebe, N., Patras, I.: Amigos: A dataset for affect, personality and mood research on individuals and groups. IEEE Transactions on Affective Computing (2018)
4. Harper, R., Southern, J.: A bayesian deep learning framework for end-to-end prediction of emotion from heartbeat. IEEE Transactions on Affective Computing (2020)
5. Jemioło, P., Storman, D., Mamica, M., Szymkowski, M., Żabicka, W., Wojtaszek-Główka, M., Ligęza, A.: Datasets for automated affect and emotion recognition from cardiovascular signals using artificial intelligence — SR. Sensors (in press) (2022)
6. Marín-Morales, J., Higuera-Trujillo, J.L., Greco, A., Guixeres, J., Llinares, C., Scilingo, E.P., Alcañiz, M., Valenza, G.: Affective computing in virtual reality: emotion recognition from brain and heartbeat dynamics using wearable sensors. Scientific reports **8**(1), 1–15 (2018)
7. Russell, J.A.: A circumplex model of affect. Journal of personality and social psychology **39**(6), 1161 (1980)

An application of U-Net for cell detection in fragments of cytological smear images

Antonina Pater[1][0000-0003-4441-2712], Łukasz Roszkowiak[1][0000-0002-1878-6588], Jakub Żak[1][0000-0003-3826-7846], Krzysztof Siemion[1,2][0000-0003-2354-9891], Anna Korzyńska[1][0000-0002-6488-4832]

[1] Nalecz Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Ks. Trojdena 4 st., 02-109 Warsaw, Poland
[2] Medical Pathomorphology Department, Medical University of Bialystok, Poland
apater@ibib.waw.pl

Abstract. Cervical cancer is one of the most lethal cancers in women, although early detection may allow successful treatment. Providing computer and automatic support during a diagnosis of cervical smears can accelerate cancer detection time as well as lower its cost. Here we show the first step towards developing such a system: A U-Net-based method for cell detection in small fragments of whole slide images of cervical pap smears. Additionally, two datasets containing 1748 and 1887 cells each were prepared and used to train and test the method. The method yields a 0.88 F1 score and will be used to develop a whole system for automatic support in cervical smear diagnosis

Keywords: biomedical engineering, digital pathology, image detection.

1 Introduction

Cervical cancer is the fourth most common cause of death from cancer in women [1]. However, prevention in the form of evaluation of cytological smears, executed every 3-5 years (depending on a type of smear) can reduce the incidence rate up to 60% [2].

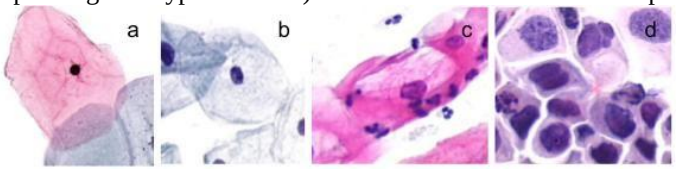


Fig. 1. Examples of normal cells (a and b) and abnormal cells (c and b).

There are two main categories of abnormal epithelial squamous cells (see Fig. 1) found in gynecological cytology smears, low-grade squamous intraepithelial lesions (LSIL) and high-grade squamous intraepithelial lesions (HSIL). Atypical cells are classified into atypical squamous cells of undetermined significance (ASC-US) or atypical squamous cells – cannot exclude HSIL (ASC-H) [3]. The treatment following

a positive result depends on the category of detected abnormal or atypical cells. If LSIL or ASC-US cells are found, an HPV test is done or another smear should be collected after 6 months. If HSIL or ASC-H cells are found, a biopsy should be performed to diagnose the patient [2].
Cytology evaluation is performed manually by highly trained cyto-diagnosticians and physicians and is very time-consuming, therefore numerous machine learning (ML) methods have been developed to support and accelerate the process, the oldest being described in 1981 [4]. A review of the ML methods developed until 2019 can be found in article [4]. Few of ML methods focuse on single category of cells, for example Tao et al. focused on distinguishing high-risk slides form ASC-US slides [5]. This work presents the results of U-Net [6] trained to perform a detection task on a newly-made set of cytological smear WSI fragments (tiles) and evaluation of the task on the testing set using F1 score. The method described in this research is a part of a system of methods being developed for the analysis of pap smears.

2 Materials

Specimens used in this research were collected in a private clinic *Arsmedica* in Bialystok in accordance with a written agreement between the clinic and IBBE, along with ethic permission. The 20 slides were digitized and annotated by cyto-diagnosticians and physicians. Annotations were in a form of color-coded dots, each color symbolized a different cell type. Two sets - training and testing were extracted from the collection.
The training dataset is made of tiles that were chosen and annotated by cyto-diagnosticians in a form of color-coded marks on WSI tiles. Among these tiles, 50 were chosen that contained low-grade squamous intraepithelial lesion (LSIL) or high-grade squamous intraepithelial lesion (HSIL) cells. Tiles, each sized 3500x3500 pixels were cut into 4 smaller fragments and compressed to size 512x512 pixels to meet usual neural networks size requirements. Based on such images ground truth maps were prepared.
Areas of cells belonging to the same group were marked on separate black and white maps using GIMP [7]. An example of cell maps is shown in Fig. 3. In this study, the cell maps were summed to allow differentiation between cell areas and background with noise. 200 tiles were prepared, each containing around 15 cells each. A summary of the types of cells included in the training set is shown in Fig 2.
The testing set consists of 60 images containing approx. 33 cells each, yielding 1887 cells and cell clusters in total. Images were cut out from 20 whole slide images (WSI) and were selected to show a large number of distinguishable cells and diverse backgrounds.

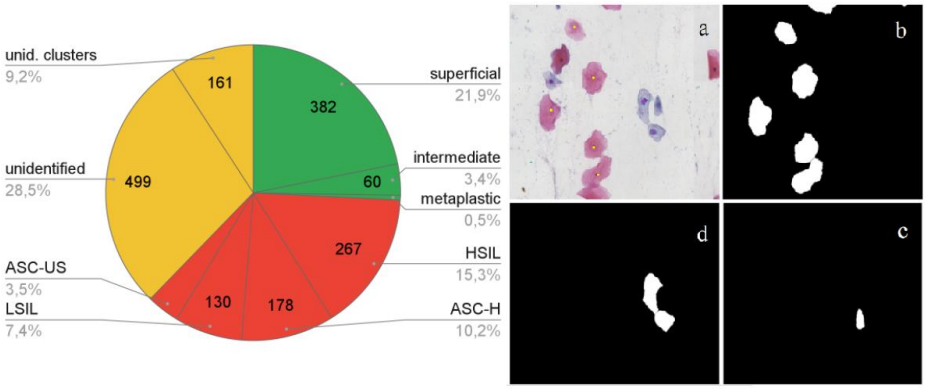


Fig 2. Left: percentages and numbers of cells representing different types in the training set of images. The total number of cells: 1748. Right: marked WSI tile a) and maps derived from it: b) superficial squamous cells, c) ASC-US, d) LSIL.

Cells and cell clusters were annotated by bounding boxes using software called VGG Image Annotator [8]. An example of an annotated image is shown in Fig. 2. Fragments of cells visible on the edges of the tiles were excluded from the annotations since such fragments do not contain full information about the cells.

3 Methods

A convolutional neural network called U-Net (with Inception Network [9] used as a backbone) was trained to perform detection of cell areas. The network's parameters are shown in table 1. The network was pretrained on ImageNet [10] dataset. Transfer learning was done using the training set discussed earlier. Network was tested with the testing set. Then, using the network's output and bounding boxes as a reference, an automatic method for preparing data for the process of evaluation of the method's performance was developed.

Table 1. Parameters of the neural network used in the study.

Parameter	Steps	Epochs	Batch size	Learning rate	Decay	Loss function	Metric function
Value	200	40	4	10^{-6}	10^{-8}	Binary cross-entropy	Binary accuracy

The evaluation method consisted of 3 steps performed on each image:

1. Finding contours in the output of interference with Otsu thresholding and filling them in white.
2. Dilating the white areas and enclosing them with a box.
3. Finding centers of annotating bounding boxes and marking them with a black dot.

If a dot exists on a white background it is treated as a true positive, a dot on a grey background is a false negative, box without a dot symbolizes a false positive.

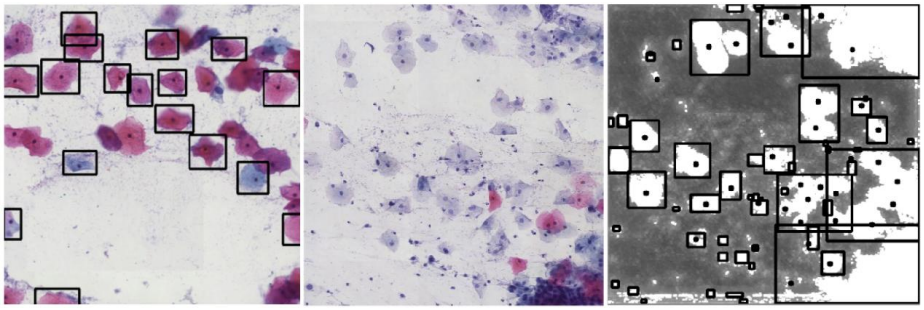


Fig. 3. An example of a bounding box annotation done only on non-clustered cells (left). A tile (middle) and its interference with evaluation marks (right). Dots mark the true position of cells (or cell clusters), white areas - network answers, boxes enclose white areas, serving as an estimation of supposed cell placements.

4 Results and discussion

The interference results were evaluated using the verification method discussed earlier, statistical measures are shown in table 2. Kurnianingsih et al [4] reached similar recall of 0.91 and slightly higher precision (0.92) and F1 (0.91) in their segmentation study. It is clear that our method shows high sensitivity, allowing it to be used in further steps towards developing a cell differentiating method.

Table 2. Summary of measures used to evaluate the network's results.

Measure	All cells	True positives	False negatives	False positives	Sensitivity	Precision	F1 score
Value	1887	1716	171	308	0.91	0.85	0.88

5 Conclusion

Cytology of the uterine cervix is a common screening method, allowing to detect precancerous changes, thus saving women from developing hard-to-treat cancer. The process of developing automation tools for cytology diagnosis is still not finished, however, neural networks seem to be a crucial method of support for physician diagnosis, allowing faster evaluation of specimens. The proposed study focuses on developing the first stage of a method for supporting cytological smear evaluation. The output of this method will be used in the next stage of differentiating normal cells from lesions. Future plans also include use of color normalization and advanced augmentation methods.

References

1. B. Stewart, World cancer report 2014, Lyon, France Geneva, Switzerland: International Agency for Research on Cancer,Distributed by WHO Press, World Health Organization, 2014.
2. M. Arbyn, A. Anttila, J. Jordan, G. Ronco, U. Schenck, N. Segnan, H. Wiener, A. Herbert and L. von Karsa, „European Guidelines for Quality Assurance in Cervical Cancer Screening. Second Edition—Summary Document," Annals of Oncology, tom 21, p. 448–458, March 2010.
3. L. Frappart, Histopathology and cytopathology of the uterine cervix = Histopathologie et cytopathologie du col utérin : digital atlas, Lyon: IARCPress, 2004.
4. Kurnianingsih, K. H. S. Allehaibi, L. E. Nugroho, Widyawan, L. Lazuardi, A. S. Prabuwono and T. Mantoro, „Segmentation and Classification of Cervical Cells Using Deep Learning," IEEE Access, tom 7, p. 116925–116941, 2019.
5. X. Tao, X. Chu, B. Guo, Q. Pan, S. Ji, W. Lou, C. Lv, G. Xie and K. Hua, „Scrutinizing high-risk patients from ASC-US cytology via a deep learning model," Cancer Cytopathology, March 2022.
6. O. Ronneberger, P. Fischer and T. Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015.
7. The GIMP Development Team, GIMP, 2019.
8. A. Dutta, A. Gupta and A. Zissermann, VGG Image Annotator (VIA), 2016.
9. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, „Rethinking the Inception Architecture for Computer Vision," December 2015.
10. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, „Imagenet: A large-scale hierarchical image database," w 2009 IEEE conference on computer vision and pattern recognition, 2009.

Building an injury prediction model using football dataset with unbalanced classes

Aleksandra Sadurska[1] and Tomasz Piłka[2]

[1] Faculty of Mathematics and Computer Science
Adam Mickiewicz University in Poznań, Poland
alesad3@st.amu.edu.pl

[2] Faculty of Mathematics and Computer Science
Adam Mickiewicz University in Poznań, Poland
tomasz.pilka@amu.edu.pl

Abstract: Every year we observe growing expectations towards the performance of professional football players. Increasing overall player load, leads to higher risk of occurrence of an injury. Injuries have a significant impact not only on football team performance but also on club budget. The objective of this research is to present results of injury prediction models implemented with various machine learning algorithms and sampling methods. The dataset and results were compared with a similar study, to comprehend whether the collected dataset is sufficient to predict an injury with certain precision. Combination of XGBoost algorithm and ROSE sampling method gave the best prediction results (F-Measure: 0.40, Precision: 0.25, Recall: 1.0).

Keywords: Football; Injury Prediction; Sports Analytics; Unbalanced Data

1 Introduction

1.1 Why is injury prediction important?

Every year we observe growing expectations towards the performance of professional football players. Increasing the number of high intensity runs, accelerations, decelerations and in the result overall player load, leads to higher risk of occurrence of an injury [1]. Injuries have a significant impact not only on football team performance but also on club budget. According to research made in English Premier League a team loses an average of £45 million due to injury-related decrement in performance per season [2].

1.2 Research objectives

The objective of this research is to present results of injury prediction models implemented with various machine learning algorithms and sampling methods. This kind of experiment has been done before, for example by Bruno Goncalo Pires Martins in his study [3]. However, in the aforementioned research, models were trained on dataset collected from another league and with additional parameters like

heart rate or rate of perceived exertion. Later in this article, I will compare the dataset and results used in both experiments, to comprehend whether the collected dataset is sufficient to predict an injury with certain precision.

1.3 What kind of injury can we predict?

Football is a high intensity sport with many interactions between players, like headings or tackles. Many of them may cause injuries that are clearly impossible to predict. On the other hand, just as common are non-contact injuries, usually resulting from over- or undertraining. They are caused by too high or too low player load within a certain timeframe, which depends on the applied training periodization model.

2 Dataset

2.1 Data characteristics

The data were collected during pre-season and the first round in the professional football league, among 31 players with an average age of 25. The data were collected using Catapult wearable global positioning trackers, both during training and match activities. Dataset used in this research contains information about 2733 events, described by the following parameters: High Speed Running, Maximum Velocity, Running, Velocity Band Total Effort Count, Sprint, Total Player Load, Field Time, Player Load Per Minute, Accelerations, Decelerations.

The dataset consists only of external load parameters, whereas in the study [3], the parameters of internal load, such as heart rate exertion, energy expenditure or rate of perceived exertion were also taken into account.

2.2 Unbalanced classes problem

Characteristics of this study are associated with unbalanced classes problem. From over 2700 events in the dataset, only 28 of them are labeled as an injury, so the ratio between classes is about 99:1. Due to uneven classes distribution, after splitting dataset into train and test set, following oversampling methods have been applied to trainset to balance both classes sample size: Synthetic Minority Oversampling Technique, Random Over-Sampling Example and Adaptive Synthetic [3].

3 Machine learning

3.1 Implemented machine learning algorithms

Machine learning methods of different characteristics and complexity were compared. The models were implemented using the following algorithms: Naïve Bayes, Support Vectors Machine, Random Forest, AdaBoost, XGBoost [7, 8].

3.2 Model evaluation

The most popular evaluation metric to assess the performance of a machine learning model is accuracy [4], which can be considered as the probability of identifying the right class of an observation. Nevertheless, on an unbalanced dataset accuracy can easily achieve 99% without correctly classifying any of the target examples. Therefore, it is not an appropriate method of evaluating an injury prediction model.

In this case it is more accurate to use precision, recall and F1-measure metrics [5]. From a perspective of injury prediction, the exclusion of false positives is most expected, because potentially stopping a player in a particular match with no reason not only can cost the team a points, but also trust for predictive analysis.

3.3 Results

Results for each model and balancing method are presented in Table 1. Each combination is described by F-measure, precision and recall evaluation metrics. Prediction was done on a test collection of 543 non-injury and 4 injury events .

Table 1. Results for each balancing method and model

Balancing	Model	F-Measure	Precision	Recall
ROSE	Naïve Bayes	0.02	1.0	0.01
ROSE	SVM	0.04	0.5	0.02
ROSE	Random Forest	0.10	0.5	0.06
ROSE	AdaBoost	0.15	0.75	0.08
ROSE	**XGBoost**	**0.40**	**0.25**	**1.0**
ADASYN	Naïve Bayes	0.02	0.75	0.01
ADASYN	SVM	0.04	0.5	0.02
ADASYN	Random Forest	0.04	0.25	0.02
ADASYN	AdaBoost	0.05	0.5	0.03
ADASYN	XGBoost	0.15	0.25	0.11
SMOTE	Naïve Bayes	0.02	0.75	0.1
SMOTE	SVM	0.04	0.5	0.02
SMOTE	Random Forest	0.04	0.25	0.02
SMOTE	AdaBoost	0.05	0.5	0.03
SMOTE	XGBoost	0.17	0.25	0.12

Combination of XGBoost algorithm and ROSE sampling method gave the best prediction results (F-Measure: 0.40, Precision: 0.25, Recall: 1.0). There is still room for improvement, but what is important, is that there are no non-injury events classified as injury, so no false positive results. In the confusion matrix below (Fig. 1), distribution of correctly and incorrectly classified items is shown.

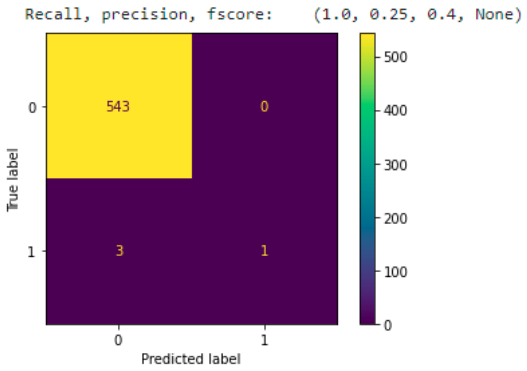


Fig 1. Confusion matrix for XGBoost algorithm and ROSE sampling method

In study [3], despite differences in dataset structure, XGBoost combined with the ROSE sampling method also gave the best prediction results (F-Measure: 0.22, Precision: 0.13, Recall: 0.67).

4 Conclusions

The results of both studies give a good perspective for further work, but in order to consider the prediction model ready for use in practice, its effectiveness should be improved. According to research conducted so far there are plenty of possibilities to achieve this goal. In the study [6], authors presented a multi-dimensional model to predict whether a player will get injured or not. Mentioned model uses lots more and also more complex parameters, like Acute:Chronic Workload Ratio, Exponential Weighted Moving Average or previous injury. Extending the dataset with additional features, like external factors related with training or internal factors, such as heart rate during activity could improve model effectiveness. Nevertheless, what should be pointed out is that injury prediction is not a fully explored problem yet. Due to the complexity of the problem, there is no single confirmed way to improve the model results.

References

1. Bengtsson, H, Ekstrand, J, Hägglund, M. Muscle injury rates in professional football increase with fixture congestion: an 11-year follow-up of the UEFA Champions League injury study. Br J Sports Med. 2013;47(12):743–747.
2. Eliakim E, Morgulev E, Lidor R, et al Estimation of injury costs: financial damage of English Premier League teams' underachievement due to injuries; BMJ Open Sport & Exercise Medicine 2020;6:e000675. doi: 10.1136/bmjsem-2019-000675
3. Martins, B. Predicting the risk of injury of professional football players with machine learning.
4. Provost, F., Fawcett, T., Kohavi, R. The Case Against Accuracy Estimation for Comparing Induction Algorithms. Proceedings of the Fifteenth International Conference on Machine Learning, 445–453. https://doi.org/10.1.1.49.5218

5. Hossin M, Md Nasir S. A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process 5.2 (2015): 1.
6. Rossi A., Pappalardo L., Cintia P., Iaia F., Fernández J., Medina D. Effective injury forecasting in soccer with GPS training data and machine learning. PLOS ONE. 13. e0201264. 10.1371/journal.pone.0201264.
7. Soofi A. A., Awan A. Classification Techniques in Machine Learning: Applications and Issues, 459-465. https://doi.org/10.6000/1927-5129.2017.13.76
8. Sagi O, Rokach L. Ensemble learning: A survey. WIREs Data Mining Knowl Discov. 2018; 8:e1249. https://doi.org/10.1002/widm.1249

Can reinforcement learning agents be e-teachers? Application of machine learning in education

Marcin Andrzejewski[1], Aleksander Iżemski[1], Piotr Linkowski[1], Kamil Pluciński[1], Eryk Szpotański[1], Mateusz Lango[1], and Krzysztof Jedrzejewski[2]

[1] Institute of Computer Science, Poznan University of Technology, Poznan, Poland
{marcin.m.andrzejewski,aleksander.izemski,
piotr.m.linkowski,eryk.szpotanski}@student.put.poznan.pl
[2] AI Learning Capabilities, Efficacy & Learning, Pearson
krzysztof.jedrzejewski@pearson.com

Abstract. E-learning systems are ubiquitous across the whole world. These systems depend on teachers who manually assign tasks to be solved by students in order to optimally develop their skills. However, this is possible only when a teacher runs a course with a small number of students. In large groups teachers are not able to track learning needs of each student. Supporting human teachers with machine learning-based recommendations may allow providing fully personalized learning experience regardless of course size. This paper investigates the possibility of using reinforcement learning algorithms to provide such support. We prepared an environment simulating an e-learning platform. Then we performed an experiment, in which a reinforcement learning agent sets tasks to students to develop their skills. Our experiments show that machine learning algorithms are able to efficiently teach students to increase their level of skill proficiency.

Keywords: reinforcement learning · education · e-learning

1 Introduction

Education is a key element of both personal and economic growth. Better education correlates with higher weekly earnings, a lower unemployment rate, and with the reduction of child mortality in developing countries. Therefore, research towards more effective, personalized learning methods is of particular interest.

Recently, e-learning systems become increasingly popular as they offer multiple benefits such as access from any place with an Internet connection and significantly lower costs compared to traditional learning. Furthermore, e-learning opens possibilities of improving the learning experience by constructing systems adaptively selecting content tailored for the needs of a particular student. Especially, the use of reinforcement learning (RL) to discover better teaching policies can result in much more effective usage of students' time, enabling them to make faster progress and focus their efforts on the most needed learning content.

The idea of individualizing the education process with RL dates back to 1960 [1]. Since then, many different solutions have been proposed. For instance,

many works explore the framework of multi-armed bandits to learn the most suitable task for the student. This also includes techniques based on contextual multi-armed bandits exploiting learning data from previous similar students. Another popular approach is to train policies that can provide hints as feedback and scaffolding. The next group of methods applies reinforcement learning for modeling students (i.e. agents) to, e.g., diagnosing students' mistakes based on their attempted solutions. Recently, many works have been conducted on NeurIPS 2020 Education Challenge task, where the algorithms have to estimate the student's knowledge based on a limited number of multiple-choice questions.

This paper investigates the possibility of applying reinforcement learning methods to find a teacher policy for an interactive e-learning platform that maximizes learning outcomes. The solved task is not limited to estimating the students' current knowledge but includes simultaneously estimating and improving students' knowledge. The paper describes the design of an e-learning platform simulator based on a psychometric model with a proper interface for training RL algorithms. The design of other such RL simulators like OpenAI Gym or ViZDoom resulted in bigger research interest and novel algorithms. Another contribution of this paper is the experimental evaluation of several modern reinforcement learning methods on the proposed simulated e-learning platform.

2 Our approach

In this part, we cast the learning process at an e-learning platform to the reinforcement learning framework and describe our implementation of it.

An episode in the environment mimics a student's individual learning process that ends with an exam with previously unknown questions. The agent, which can be understood as a teacher, interacts with the environment by selecting the tasks that the student will attempt to solve. The task is characterized[3] by particular skill needed to solve it (learning objective) and the difficulty level on each skill. The student can solve the task correctly or incorrectly (which also includes the lack of answer), which is reflected in the environment's state, producing feedback for the agent. In both cases, i.e. incorrect/correct answer, the student's proficiency in a skill increases by a certain degree, as even unsuccessful attempts require some cognitive effort and reflection on the use of a given skill. A task can be potentially solved by a student multiple times in the whole learning process. However, in such a case, every attempt is assumed to deal with another task from a group of tasks having same characteristics.

The simulation time is discrete and divided into episodes. Each episode has two phases: learning and evaluation (exam). During the learning phase, the student solves tasks iteratively selected by the teacher. Then, in the exam phase, the student receives a set of tasks to solve. The reward is always 0, excluding the exam phase, when the reward is equal to the number of correctly solved tasks.

[3] The characteristics of tasks are used for underlying computations, and are unknown both to the student and teacher.

The agent's goal is to maximize the number of correctly solved tasks during the exam, which is correlated with the student's skill development.

To model the student's interactions with tasks belonging to each skill, the popular Rasch psychometric model [3] was used. This model describes a student with a continuous proficiency. This value is on a logit scale and usually ranges from -3 to 3. It determines the probability of correctly solving an avg. difficulty task. We used a separate proficiency value for each skill.

In our implementation, the observations returned by the environment mimic a student's grade book and include how many times a given task was solved correctly and incorrectly. More concretely, it is represented as a list with a length equal to the number of possible tasks to choose from, and at position i, it contains tuples of two numbers representing correct and incorrect solutions of i-th task.

3 Experiments

The purpose of the experiments was to investigate the performance of selected RL algorithms, namely A2C (Advantage Actor Critic) [2] and PPO (Proximal Policy Optimization) [4], on the implemented e-learning platform and to compare them to a baseline selecting actions randomly.

Each agent was run for 25,000 episodes. During each episode, the new student was learning s skills by solving tasks (actions) selected by the teacher (agent). For each skill and each of the seven difficulty levels, one task associated with them is available on the e-learning platform. In the case of 7 skills, it means 49 possible tasks to be chosen by the agent. Student proficiencies are sampled from a normal distribution with a mean of 0 and a standard deviation of 1/3, clipped to a range from -1 to 1. At the end of each episode, the student took an exam that consisted of 2 difficult tasks (at the level 6 out of 7) for each skill developed. Experiments with s equal to 1, 3, 5 and 7 have been performed. The length of learning phase was set to $10s$. For example, an episode of teaching 5 skills allows for 50 interactions with tasks.

The goal of the system is to improve the student's proficiency at each skill, so the system was evaluated using the mean of proficiencies. Each parameter configuration was run 5 times with different random seeds[4]. The results of the experiments are presented in Figure 1. The plot on the left shows the mean of student's proficiencies after the exam, when the number of developing skills was set to 5. One can notice here that reinforcement learning approaches are considerably better than the baseline. This means that the agent is able to identify which task support the development of which skill, and based on the observation, estimate which skills the student is deficient in and should improve.

On the other hand, the plot on the right presents mean proficiency of the student after the 25000 episode depending on the number of developing skills and used RL algorithm. As the number of skills increases, the number of possible tasks to choose from increases too. We can observe, that in the result, it becomes

[4] The source code for the experiments can be found in `https://github.com/er713/Schooling-RL`, commit hash: `3f1db56`

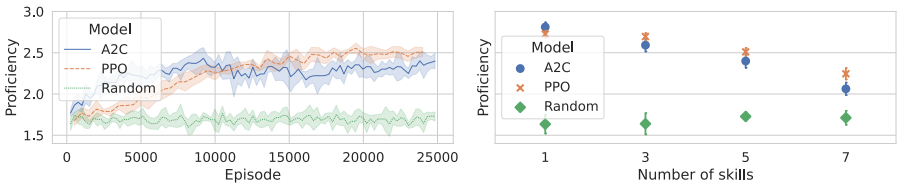


Fig. 1: The figure presents the experimental results. On the left, the difference in performance between algorithms is shown, when the number of skills to learn by the student was set to 5. On the right, the performance of the algorithms after 25000 episodes is compared against the increasing difficulty of the environment (the number of skills and the action space increases). Both plots show the mean values of the 5 runs with different seed values, along with the standard deviation.

noticeably more difficult for the agent to properly map tasks to skills development of which they support. It negatively affects the proficiency the student is able to achieve in the end. Furthermore, we can see that in this limited time, the PPO is able to find the appropriate strategy to teach the student faster.

4 Summary

This paper investigated whether it is possible to apply RL methods to teach students on an e-learning platform using personalized, adaptive study plans. We have shown that algorithms we experimented on can learn development of which skill is supported by each task and how to select tasks appropriate for each student needs. Following that, reinforcement learning teacher successfully improved the student's proficiency. In reference to these results we demonstrated that reinforcement learning agents can support learning in e-learning platforms. As part of our future work, we plan to extend our approach using other algorithms (e.g. Deep-Q Network) or methods based on Monte Carlo Tree search. Moreover, we plan to extend the prepared environment by adding new student psychometric models like a Cognitive Diagnostic Models or Bayesian Knowledge Tracing.

Ack. Authors were supported by AI Tech project sponsored from POPC programme POPC.03.02.00-00-0001/20

References

1. Doroudi, S., Aleven, V., Brunskill, E.: Where's the reward? International Journal of Artificial Intelligence in Education **29** (2019)
2. Mnih, V., Badia, A.P., Mirza, M., et al.: Asynchronous methods for deep reinforcement learning. In: ICML. p. 1928–1937 (2016)
3. Rasch, G.: Probabilistic models for some intelligence and attainment tests. Danish Institute for Educational Research (1960)
4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017), `http://arxiv.org/abs/1707.06347`

COVID-19 diagnosis based on chest X-ray images

Agata Giełczyk[1][0000−0002−5630−7461], Anna Marciniak[1,2][0000−0002−9900−3951], and Martyna Tarczewska[1]

[1] University of Science and Technology, Bydgoszcz, Poland
agata.gielczyk@pbs.edu.pl
[2] Faculty of Medicine Ludwik Rydygier Collegium Medicum in Bydgoszcz, Nicolaus Copernicus University in Torun, Bydgoszcz, Poland

Abstract. The SARS-CoV-2 pandemic began in early 2020 and paralyzed human life all over the world. Due to the overstretching of the healthcare system, the rapid and efficient tools for COVID-19 diagnosis and distinguishing became especially needed. In this article we implement a ML-based approach with some pre-processing methods, and verify whether they can improve the COVID-19 positive vs. negative classification using the chest X-ray images. We provide also some promising results: Accuracy $> 97\%$, Precision $> 96\%$, Recall $> 96\%$ and F1-score $> 96\%$.

Keywords: SARS-CoV-2 · COVID-19 · Machine learning · Image processing · Pre-processing

1 Introduction

The beginning of the COVID-19 pandemic in 2020 has shaken up the modern world. It has caused societies to close, crowded streets to become deserted, pubs and clubs to be silenced, and popular meeting places to die down. Currently, people all over the world are doing their best to overcome the pandemic's impact on the social, medical, psychological, economic, and industrial aspects of society. In this moment, the reverse transcription polymerase chain reaction (RT-PCR) testing is the main screening method for detecting COVID-19 infections. However, patients suffering from COVID-19 can also present with abnormalities on chest X-ray (CXR) images that are characteristic of infection [5]. This imaging modality is highly available and accessible in many clinical locations, and it is considered standard equipment in most healthcare systems. Moreover, CXR imaging is more widely available than CT (computed tomography) imaging, especially in developing countries due to high equipment and maintenance costs. X-ray analysis can be time-consuming and requires highly educated specialists to interpret. But, the use of machine learning (ML)-based methods can improve efficiency, support medics in the diagnosis of COVID-19, speed up the time to diagnosis, and lighten the already burdened health care system.

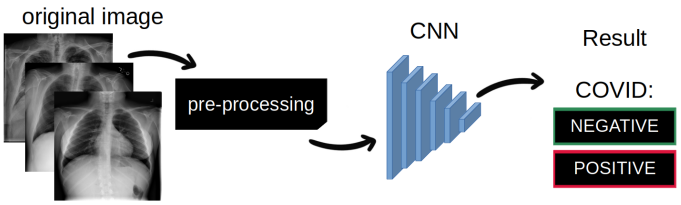
In this article we propose a ML-based method for COVID-19 infection detection using chest X-ray images analysis with different pre-processing methods. The presented article will be possibly extended by adding more lung diseases e.g., pneumonia.

The intensification of the COVID-19 pandemic caused that more and more researchers became working on the automatic disease recognition. Some of them focus strictly on the image pre-processing. It has numerous advantages: reduces noises, enhance image and by resizing image can reduce the computation time. In [6, 7] the segmentation problem was raised as an emerging issue. Medical data suffers from weak annotations, as the description of the CT and X-ray images is complicated and time consuming. Authors in [2] proposed a novel, hybrid, multimodal deep learning system. With the use of Contrast-Limited Adaptive Histogram Equalization (CLAHE) and a Butterworth bandpass filter, the authors were able to enhance the contrast of X-ray images and eliminate the noise leading to an accuracy of 99.93%. Mahdy et al. in [4], proposed a method to automatically classify COVID-19 chest X-rays using a multi-level threshold based on the Otsu algorithm and support vector machine (SVM).

2 Proposed method

In this research, we used the dataset available for the public at [1]. It consists of images collected from various sources: GitHub repository, Kaggle, Radiopedia, Italian Society of Radiology (SIRM), and Figshare data repository websites. The dataset contains almost 7.000 samples presenting posteroanterior (PA) chest X-ray images.

Fig. 1. The overview of the proposed method: original CXR image, black box of pre-processing, CNN and the final result



To verify how the selected pre-processing method affects the final classification result, we proposed a baseline system. The general overview of this system is presented in Fig. 1. The black box visible in Fig. 1 marks the selected pre-processing method. The pre-processing step is an important element in the image analysis schema. It can enhance the original image and reduce noise or unwanted details. In our research, we examined 6 different approaches to pre-processing:

1. None — in this method the only one element is size reduction.

2. Histogram equalization — this method extends the pixel's intensity range from the original range from 0 to 255. As the result, the enhanced image has a wider range of intensity and slightly higher contrast.
3. Hist. eq. + Gaussian blur — this filter reduces some noise and unwanted details that can be confusing for the neural network; the filter kernel size was experimentally set to 5×5 size.
4. Hist. eq. + bilateral filter — this filter also reduces some noise and unwanted details that can be confusing for the neural network, but its main feature is to preserve edges; the experimentally set up parameters of the filter: $diameter = 5$, $\sigma_{color} = \sigma_{space} = 75$.
5. Adaptive masking — in [3] the authors proved that by removing the diaphragm from the sample it is possible to improve the classification results. In this proposed pre-processing method, we first found the maximum (max) and minimum (min) intensity of pixels and then applied the binary thresholding using the threshold expressed in Eq. 1. The next step used morphologic closing. This creates the adaptive mask that after bitwise operation removes the diaphragm from the source image.
6. Adaptive masking + hist. eq. + Gaussian blur — this method joins adaptive masking with histogram equalization and Gaussian blur (with kernel size 5×5 again.

$$threshold = min + 0.9 \cdot (max - min) \tag{1}$$

In a classification step we implemented a convolution network (CNN). It can be described as a chain of convolution layers, with rectified linear unit activation functions, pooling layers, and batch normalization operations. In this specific approach we used 12 layers. The output from the neural network is the classification, whether the patient is COVID-19 positive or negative. All the experiments were executed using the online Kaggle notebook. There were almost 7,000 samples in the dataset. We decided to divide the dataset into three disjoint subsets: training-65%, validating-15%, and testing-20%. All of the experiments were executed 3 times to prove their independence from the learning data. Thanks to the balance in the dataset, we did not need any sample augmentation.

3 Results

The above-mentioned experiments provided some promising results. We used 4 parameters for the evaluation methods - accuracy, precision, recall, and F1-score. The parameters were calculated using a confusion matrix reporting the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The obtained results are presented in Table 1.

4 Conclusion

In this article, we proposed a novel approach for the fully automated analysis of COVID-19 chest X-ray images using a convolutional neural network. We

Table 1. The obtained results for all pre-processing methods

No	Method	Accuracy	Precision	Recall	F1-score
1.	None	0.9385	0.8566	0.9806	0.9144
2.	Histogram equalization	0.9602	0.9340	0.9482	0.9411
3.	Hist. eq. + Gaussian blur	0.9609	0.9199	0.9676	0.9432
4.	Hist. eq. + Bilateral filter	0.9725	0.9436	0.9762	0.9597
5.	Adaptive masking	0.9544	0.9739	0.8877	0.9288
6.	**Ada. mask. + hist. eq. + Gauss. blur**	**0.9761**	**0.9614**	**0.9676**	**0.9645**

also presented an improvement in the proposed method by introducing the pre-processing part of the ML-based system. In this early step of image analysis, a few crucial operations are performed: adaptive masking (very light parts of the image are removed), histogram equalization (widens the range of pixels intensity), and Gaussian blur (removes noise and some unwanted details). The presented results proved that the proposed pre-processing method increases the efficiency of the system as the F1-score raised from 91% to over 96%. Our results are comparable to other similar ML-based approaches in the literature, but there are plenty of pre-processing methods that can improve the efficiency of the system and be implemented in future work. Other possible extension of this method is to add more classes and more diseases to classification.

References

1. COVID19 Pneumonia NormalChest Xray PA Dataset. www.kaggle.com/94amanullahasraf/covid19-pneumonia-normal-chest-xray-pa-dataset, [Online; accessed 03.01.2022]
2. Al-Waisy, A., Mohammed, M.A., Al-Fahdawi, S., Maashi, M., Garcia-Zapirain, B., Abdulkareem, K.H., Mostafa, S., Kumar, N.M., Le, D.N.: Covid-deepnet: hybrid multimodal deep learning system for improving covid-19 pneumonia detection in chest x-ray images. Computers, Materials and Continua **67**(2), 2409–2429 (2021)
3. Heidari, M., Mirniaharikandehei, S., Khuzani, A.Z., Danala, G., Qiu, Y., Zheng, B.: Improving the performance of cnn to predict the likelihood of covid-19 using chest x-ray images with preprocessing algorithms. International journal of medical informatics **144**, 104284 (2020)
4. Mahdy, L.N., Ezzat, K.A., Elmousalami, H.H., Ella, H.A., Hassanien, A.E.: Automatic x-ray covid-19 lung image classification system based on multi-level thresholding and support vector machine. MedRxiv pp. 2020–03 (2020)
5. Ng, M.Y., Lee, E.Y., Yang, J., Yang, F., Li, X., Wang, H., Lui, M.M.s., Lo, C.S.Y., Leung, B., Khong, P.L., et al.: Imaging profile of the covid-19 infection: radiologic findings and literature review. Radiology: Cardiothoracic Imaging **2**(1), e200034 (2020)
6. Saood, A., Hatem, I.: Covid-19 lung ct image segmentation using deep learning methods: U-net versus segnet. BMC Medical Imaging **21**(1), 1–10 (2021)
7. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Medical Image Analysis **63**, 101693 (2020)

COVID-19 Early Diagnosis with the Use of Machine Learning

Anna Marciniak[1,2][0000−0002−9900−3951], Agata Giełczyk[1][0000−0002−5630−7461], Martyna Tarczewska[1], and Sylwester Kloska[2][0000−0002−5165−9302]

[1] University of Science and Technology, Bydgoszcz, Poland
agata.gielczyk@pbs.edu.pl
[2] Faculty of Medicine Ludwik Rydygier Collegium Medicum in Bydgoszcz, Nicolaus Copernicus University in Torun, Bydgoszcz, Poland

Abstract. Since December 2019 COVID-19 disease has spread all over the world, paralyzing human life and threatening our security. Thus, the need for a novel and fast approach to diagnosing COVID-19 infections became predominant. This work proposes a machine learning based method to classify the early symptoms of SARS-CoV-2 infection. Three classifiers have been selected: Random Forest, XGBoost, and LightGBM. Additionally, each classifier was tested on an unbalanced and balanced dataset and with the use of default and tuned hyper parameters. XGBoost model gave the best results after training at balanced dataset with Accuracy=79.94%, Precision=93.32%, Recall=64.50%, and F1-score=76.62%. The selected ML model was then linked to a mobile application that contained a questionnaire about symptoms. After completing the questionnaire, the result obtained from the ML model is returned to the user in the application. The obtained results are prognostic results, suggesting social isolation and/or performing additional tests, e.g., a PCR test, and are not a substitute for the professional medical diagnosis.

Keywords: SARS-CoV-2 · COVID-19 · Machine Learning · XGBoost · Random Forest · LightGBM · Classification

1 Introduction

Since December 2019 SARS-CoV-2 virus has spread all over the world from Wuhan, China, causing a disease known as coronavirus disease COVID-19. On 30 January 2020, the World Health Organization (WHO) announced a public health emergency, and the epidemic rapidly evolved into a pandemic by March 2020, with a high number of cases in the Europe, especially in Italy [5]. The healthcare systems were overstretched and, as the result, patients had serious obstacles in receiving needed medical help on time. Thus, a rapid tool for the diagnosis support were especially needed.

In this article, we evaluate three ML-based models. Additionally, we examine models with/without dataset balancing and with/without hyper parameters

tuning. We propose also a mobile application that can be used for COVID-19 self diagnosis. The proposed application cannot substitute the professional medical diagnosis, but it can be the first tool used in triaging system at medical centres. It uses the machine learning methods for binary classification (COVID positive vs COVID negative). The article is constructed as follows. In section 2, we describe the materials and methods, in section 3 we provide the obtained results. The last section contains the conclusions and future possibilities.

2 Materials and Methods

The schema of the proposed application and the machine learning-based pipeline are presented in Fig. 1. In the top part of the figure the machine learning process can be observed. We provided the data, processed them and finally, we used them in order to develop the ML models. Then, we performed experiments and the most promising model was implemented in the mobile application. In the bottom part of the figure the use of the application is presented. Firstly, the user fulfill the questionnaire. Then, the answers are analysed by the ML-based model and the application provides the result - COVID positive or negative.

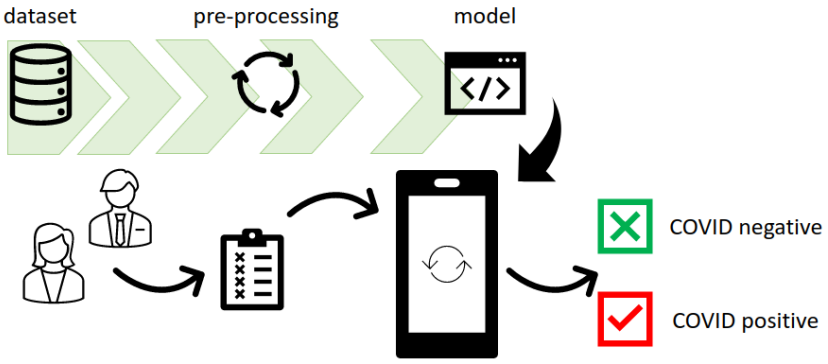


Fig. 1. The overview of the proposed method

2.1 Dataset and Data Pre-processing

In this research, we used the dataset provided by Yazeed Zaobi [7]. The dataset was created from the data gathered in Israel. It contains the following pieces of information: date of test, cough, fever, throat soreness, breath shortness, headache, age 60 and above, gender, and test indication. All negative and positive COVID-19 cases in this dataset were confirmed via RT-PCR assay.

Firstly, the column 'date of test' was removed due to its irrelevance. Secondly, all the uncertain COVID results were removed. Then, all the data were

converted into a numerical format. After the pre-processing dataset contained 2 701 378 observations. In the research we used two separable subsets with random elements: 80% of data for training and 20% for testing. Due to the significant lack of balance in the dataset (90% of observations was COVID negative), we decided to use the dataset with and without balancing. The balancing was ensured by the SMOTE algorithm [1].

2.2 Classification

In the classification step of the method we implemented and evaluated three ML-based models: Random Forest (used also in [2]), XGBoost [3] and LightGBM [4]. All models were evaluated in four different experiments - on unbalanced and balanced dataset, and with different hyper parameter values - default and tuned using the Optuna framework [6].

3 Results

The results of the conducted research are presented in Table 1. It presents the evaluation of three models by Accuracy, Precision, Recall, and F1-score. In the table, we additionally provide two time related parameters: time of learning and time of predicting. The results in time domains were evaluated using the Google Colaboratory environment. Individual classifiers results do not differ significantly for different research variants. However, for XGBoost classifier (marked in bold in Table 1) trained on default hyper parameters and balanced dataset the evaluation measures and time are the most promising, namely Accuracy=79.94%, Precision=93.32%, Recall=64.50%, and F1-score=76.62%.

Table 1. The obtained results: Accuracy, Precision, Recall, F1-score, Training time, and Prediction time for Random Forest, XGBoost, and LightGBM, both with and without balancing and hyper parameters tuning

Model	Hyper parameters	Data balance	Acc.	Prec.	Recall	F1-score	Training time[s]	Pred. time[s]
Random Forest	Default	None	0.9395	0.6489	0.5593	0.6008	109	4.68
		SMOTE	0.7993	0.9332	0.6447	0.7626	279	0.925
	Optuna	None	0.9395	0.6489	0.5594	0.6008	604	2.58
		SMOTE	0.7993	0.9332	0.6447	0.7626	844	2.61
XGBoost	Default	None	0.9386	0.6517	0.5274	0.5830	98	0.942
		SMOTE	**0.7994**	**0.9332**	**0.6450**	**0.7628**	**199**	**0.197**
	Optuna	None	0.9395	0.6489	0.5593	0.6009	857	3.37
		SMOTE	0.7994	0.9332	0.6450	0.7628	954	0.338
LightGBM	Default	None	0.9396	0.6492	0.5594	0.6010	20.3	1.79
		SMOTE	0.7994	0.9332	0.6450	0.7628	52.1	0.323
	Optuna	None	0.9396	0.6493	0.5593	0.6009	34.1	2.21
		SMOTE	0.7995	0.9325	0.6458	0.7631	55.6	0.295

4 Conclusions

Since the start of the pandemic in 2020, many new mutations of the SARS-CoV-2 virus have emerged, each with a different set of disease symptoms. This may be one of the ways of developing the prepared application - distinguishing not only sick/healthy, but also typing the virus variant. Applications of this type can potentially be used as a tool in the healthcare sector. The result obtained by analysing the survey responses by the machine learning model may help healthcare professionals in the procedure of triaging patients. However, the diagnosis obtained from a machine learning model is not a substitute for a visit to the doctor, but an indication for further diagnosis or its omission.

References

1. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
2. Gupta, V.K., Gupta, A., Kumar, D., Sardana, A.: Prediction of covid-19 confirmed, death, and cured cases in india using random forest model. Big Data Mining and Analytics **4**(2), 116–123 (2021)
3. Irawati, M.E., Zakaria, H.: Classification model for covid-19 detection through recording of cough using xgboost classifier algorithm. In: 2021 International Symposium on Electronics and Smart Devices (ISESD). pp. 1–5. IEEE (2021)
4. Kelter, D., Ghiassi, K., Patel, S., Connors, C., Bonk, M., Gray, E., Zarbiv, S., Menon, A., Juneja, P.: Use of feature engineering to predict covid-19 mortality. In: TP51. TP051 COVID: LUNG INFECTION, MULTIORGAN FAILURE, AND CARDIOVASCULAR, pp. A2630–A2630. American Thoracic Society (2021)
5. Macera, M., De Angelis, G., Sagnelli, C., Coppola, N., COVID, V., et al.: Clinical presentation of covid-19: case series and review of the literature. International journal of environmental research and public health **17**(14), 5062 (2020)
6. Srinivas, P., Katarya, R.: hyoptxg: Optuna hyper-parameter optimization framework for predicting cardiovascular disease using xgboost. Biomedical Signal Processing and Control **73**, 103456 (2022)
7. Zoabi, Y., Deri-Rozov, S., Shomron, N.: Machine learning-based prediction of covid-19 diagnosis based on symptoms. npj digital medicine **4**(1), 1–5 (2021)

Detection of Damaged Fields in AIS Messages *

Marta Szarmach[0000−0002−3793−6641]
and Ireneusz Czarnowski[0000−0003−0867−3114]

Gdynia Maritime University
81-225 Gdynia, ul. Morska 81-87, Poland
m.szarmach@we.umg.edu.pl, i.czarnowski@umg.edu.pl

Abstract. Thanks to the Automatic Identification System (AIS), a telecommunication system designed for maritime purposes, ships from a given area are able to transmit information about themselves (their movement) to other ships, so that neighboring vessels are aware of each others' trajectory and collisions between ships can be avoided. However, due to the technological limitations of AIS (such as the packet collision effect) some of the AIS data are either lost or damaged during the transmission. That is why reconstructing of AIS data is an important issue. In this paper, we propose our method for AIS data reconstruction based on machine learning techniques. Here we mainly focus on the anomaly detection phase, which leads to the identification of incorrect AIS data that has to be reconstructed. Our approach uses clustering to find abnormal datapoints and multi-label classification to distinguish which fields of AIS messages need to be corrected. The preliminary results of such anomaly detection are presented.

Keywords: AIS data reconstruction · Anomaly detection · Multi-label classification

1 Introduction

AIS (Automatic Identification System) is a telecommunication system designed for maritime purposes. It allows ships from a given area to transmit information about themselves and their trajectory to other ships, so that neighboring vessels can be aware of each others' trajectory to avoid collisions between them [5].

AIS consists of two segments. The original segment (terrestrial), utilizing very high frequency radio waves, allows only for communication in a field-of-view (up to 74 km) — ship to ship or ship to shore. To increase the communication

* The project is financed within the framework of the program of the Minister of Science and Higher Education under the name "Regional Excellence Initiative" in the years 2019 - 2022; project number 006/RID/2018/19; the amount of financing PLN 11 870 000 and partially by grant of Department of Information Systems, Gdynia Maritime University.
Our special thanks to Mr Marcin Waraksa and Prof. Jakub Montewka from Gdynia Maritime University for sharing the raw data that we used in our experiment.

range, a new, satellite segment — SAT-AIS — was introduced [3]. In SAT-AIS, a dedicated satellite mediates the communication between many terrestrial areas. However, SAT-AIS struggles againts synchonization issues, mainly packet collision effect [7]: a single satellite covers many terrestrial cells, in which the transmission is synchronised using Self-Organised Time Division Multiple Access, but the cells are not synchronised between themselves, therefore a satellite cannot properly receive data from two or more cells at the same time.

In this paper we propose a machine learning based method for reconstruction of lost or damaged AIS data. One step of the reconstruction is described, namely the anomaly detection phase, which allows identification of AIS data that requires further correction.

2 Proposed Approach of AIS Data Reconstruction

2.1 General Algorithm

The proposed approach of AIS data reconstruction can be divided into 3 steps:

1. **Step one: Clustering.** The AIS data recorded from a given area in a given time is divided into groups, such that each group consists of messages from one and only one vessel.
2. **Step two: Anomaly detection.** The aim of this phase is to identify incorrect data that requires correction: either the whole messages or their damaged parts (fields). This is done in two ways simultaneously:
   - by analysing each group obtained during the clustering phase to find outlying messages,
   - by searching for messages that form a separate group (since they differ from others so much, they might be damaged).
3. **Step three: Prediction.** The correct values of an AIS message fields that are considered damaged are predicted, based on other values from a given group.

2.2 Anomaly Detection Phase

Analysing Standalone Clusters. All messages that form separate clusters are considered outliers. Then the groups that they should belong to (consisting of messages that originated from the same ship) are found. The k Nearest Neighbours algorithm [1] is utilized here: the k-NN classifier is trained on the entire dataset (with the labels being the indices of groups that each message is assigned to) and then in predicts the right index.

Searching for Anomalies Within Clusters. In some situations, the message might not be damaged strong enough to form a standalone cluster. Hence, there is a need to analyse each group to find outlying points inside it. In our framework, the 1-D convolutional neural network is going to be used here, with its input being the waveforms of each field building a single trajectory.

Identyfing Damaged Message Fields. When a damaged AIS message is detected, the exact localization of the disturbance must be established. Since the waveform of values of a given field with a incorrect value in it somehow resembles a wavelet, a wavelet transform [2] of the preprocessed waveform is computed and $\Delta\phi$, the relative difference of its maximum value (for waveforms with and without the potentially incorrect values) is calculated. Also $\Delta\sigma$, the same difference of computed standard deviation for each waveform is calculated. The higher those differences, the more likely the field they were computed on contains an incorrect value. Then the pre-trained Random Forest [4] classifier is given the 2-element vector $[\Delta\phi, \Delta\sigma]$ and provides the multi-label classification, deciding are the differences high enough to consider the field as damaged. Since there might be disturbances in many fields in the entire message, not only one, the classification in this task must be multi-label.

3 Computational Experiment

3.1 Overview

Aim of The Experiment. The aim of the computational experiment is to verify the effectiveness of the proposed method of detecting damaged AIS messages and their fields. In this paper, we present the results of analysing only messages forming separate clusters.

Data. The data used in this experiment is a real data recorded from AIS. Among all 27 types of messages transmitted in AIS, only 3 were used — types 1, 2 and 3, called position reports, which carry the information regarding ship's trajectory [6]. The features extracted from those messages to build the input dataset are: MMSI (ship's identifier), longitude, latitude, navigational status, special manouevre indicator, speed over ground, course over ground, true heading.

The data is gathered into 3 datasets:

- 805 messages from 22 vessels from the area of Gulf of Gdańsk,
- 19999 messages from 387 vessels from the area of Baltic Sea,
- 19999 messages from 524 vessels from the area of Gibraltar.

Methodology. All 3 datasets were examined — for each, 20 randomly selected messages were artificially damaged (1 or 2 of their bits were swapped), then the anomaly detection algorithm was run to find those damaged messages (content of standalone clusters), decide which ships they were originating from (using k-NN with the $k = 5$) and then search for the damaged fields (using Random Forest classifier, $max\_depth = 5$, $n\_estimators = 15$).

3.2 Results

The mean of gathered results of each message is presented in Tab 1.

Table 1. Results of anomaly detection in AIS data in standalone clusters

-	Metric	1. dataset	2. dataset	3. dataset
1 bit damaged	Assignment	96,00%	59,00%	72,00%
	Recall	100,00%	100,00%	99,00%
	Precision	93,33%	70,83%	82,50%
	F1	95,33%	78,33%	87,33%
	Jaccard	93,33%	70,83%	82,50%
	Hamming	98,86%	93,93%	97,00%
2 bits damaged	Assignment	99,00%	68,00%	72,00%
	Recall	81,00%	79,00%	75,50%
	Precision	98,50%	82,00%	86,33%
	F1	86,53%	75,37%	76,13%
	Jaccard	79,92%	64,08%	65,42%
	Hamming	97,00%	93,36%	94,00%

The results are promising — as can be noticed, the recall (which is considered the most impactful metric here, since it is more important to maximize the detection rate of all damaged fields than to minimize the false negative rate) did not drop below 75.5%, mostly varying between 80%-100%. Naturally, the recall was slightly lower for bigger error rate and dataset size.

4 Conclusions

The proposed method of finding damaged AIS messages and their fields seems to provide promising results. However, the work is still in progress — in the near future, the effectiveness of finding anomalies inside groups should be examined, as well as the impact of batch size on the anomaly detection phase quality.

References

1. Altman, N.S.: An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician **46**(3), 175-185 (1992)
2. Debnath, L.: Wavelet transform and their applications. PINSA-A **64,A** (6), 685-713 (1998)
3. European Space Agency Homepage, https://artes.esa.int/sat-ais/overview. Last accessed 25 Feb 2022
4. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Anaysis and Machine Intelligence **20**(8), 832-844 (1998)
5. International Maritime Organization Homepage, https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx. Last accessed 25 Feb 2022
6. Recommendation ITU-R M.1371-5, https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-5-201402-I!!PDF-E.pdf. Last accessed 25 Feb 2022
7. Wawrzaszek, R., Waraksa, M., Kalarus, M., Juchnikowski, G., Gorski, T.: Detection and Decoding of AIS Navigation Messages by a Low Earth Orbit Satellite. In: Sasiadek, J. (eds) Aerospace Robotics III. GeoPlanet: Earth and Planetary Sciences, pp. 45-62, Springer, Cham (2019)

Discovering Communities in Organizational Social Network through Hierarchical Clustering

Paweł Szyman[0000−0003−3353−5522] and Dariusz Barbucha[0000−0001−7906−0740]

Department of Information Systems
Gdynia Maritime University
Morska 83, 81-225 Gdynia, Poland
p.szyman@wznj.umg.edu.pl, d.barbucha@umg.edu.pl

Abstract. Organizational social network is a kind of social network where people or units of an organization are represented in the form of vertices, and relationships or information flow between these people or units are represented in the form of edges between the vertices. By applying a methodology of social network analysis it is possible to analyze different properties of such network. The paper focuses on organizational social network of public organization located in Poland, where the structure of the network is based on digital forms of communication between people within an organization. The main contribution of the paper is to detect communities in such constructed network by applying dedicated hierarchical clustering method. The results of the computational experiment allowed one to validate detected communities and compare them with real structure of the organization.

Keywords: Organizational social network · Community detection · Hierarchical clustering · Similarity

1 Introduction

Social network analysis offers a number of interesting approaches to the analysis of different aspects of an organization and its activity. It assumes that people or units of the organization are represented in the form of vertices in the network, and relationships or information flow between these people or units are represented in the form of edges between the vertices. Although an organizational social network may have different forms depending on a source and a form of information used to its construction, one of the most representative is a social network based on different digital forms of business communication within an organization, especially using electronic mail channel. In such network the edge between vertices represents interaction in terms of messages sent or received [3].

This paper focuses on the organizational social network of one of the public organizations, which has been constructed by extracting the anonymized data referring to email communication between persons employed in this organization. The main goal of the paper is to detect a number of cohesive communities in the network, where a community can be generally defined as a group of vertices

similar to each other [5]. As a part of the analysis, the resulting communities have been compared with real structure of the organization.

The paper is organized as follows. Section 2 defines a problem of community detection and presents a hierarchical approach to solve the community detection problem in organizational social network. Section 3 presents the results of computational experiment carried out to validate the proposed approach. Section 4 concludes the paper and presents directions of the research planned in the near future.

2 Community detection in organizational social network

2.1 Community detection problem

Let $G = (V, E)$ be an undirected graph, representing a social network, where V represents a set of vertices, and E - set of edges between vertices. A subset of vertices within the graph $C \subset G$ such that connections between the vertices are denser than connections with the rest of the network is called *community*. The problem of *community detection* aims at dividing the vertices of a network into some number k of groups, while maximizing the number of edges inside these groups and minimizing the number of edges established between vertices in different groups [4]. Detailed review of the methods used to solve the problem of community detection, the reader can find for example in [5].

2.2 Hierarchical clustering approach to the community detection in organizational social network

The proposed approach (HC) to solve the community detection problem is based on agglomerative hierarchical clustering [5]. The starting point of any hierarchical clustering method is the definition of the similarity measure between vertices. In case of social networks the similarity must be inferred from the adjacency relationships between vertices. In the proposed approach the similarity measures the overlap between the vertices $i, j \in V$ given by the ratio between the intersection and the union of the neighborhoods, i.e. $s_J(i,j) = |\Gamma(i) \cap \Gamma(j)|/|\Gamma(j) \cup \Gamma(j)|$, where $\Gamma(i)$ and $\Gamma(j)$ denote the set of neighbors of vertices i and j, respectively. As a consequence, having the similarity calculated for each pair of vertices, no matter if they are connected or not, a new similarity matrix S_J is created. Due to the fact that proposed hierarchical clustering approach belongs to the agglomerative algorithms group, it has been also essential to define a measure that estimates how similar clusters are, out of the similarity matrix S_J. After a preliminary tests with different measures, the average linkage has been adapted.

3 Computational experiment

Computational experiment has been carried out in order to discover communities in the social network based on e-mail communication between employees

of the public institution through hierarchical clustering and compare detected communities with the organizational structure of the institution.

The experiment was divided into four steps. The first step was to select the observation period (six months in the experiment), a number of investigated departments (single department with the biggest number of employees - 91) and then extraction and collection of the data from organization's e-mail server logs. Total number of identified messages between the employees was equal to 10 364.

The second step was focused on building a network, where employees were represented by vertices and communication between them - by edges. It has been assumed that an edge between two vertices exists if two employees represented by these vertices exchanged at least a single message within observed period. In order to ensure the security of employees' personal data, each e-mail address was anonymized by assigning to it an individual number from 1 to 91.

The third step focused on applying the proposed hierarchical approach to discover communities in the social network built in the previous step. The proposed algorithm has been implemented using *R* software package with the *iGraph* and *LSA* libraries. In the fourth step, the obtained results (presented in Table 1 and Fig. 1) were analyzed and the communities detected in the network by HC algorithm were compared with the structure of the organization.

The result of hierarchical clustering in the form of a dendrogram is presented in Fig. 1. One can observe that 5 communities were detected in the network. In order to confirm this observation, it has been also decided to calculate the value of the silhouette index for different number of clusters. The best value (0.3265) has been obtained for 5 clusters. What is important, these results correctly refer to real organizational structure, where 5 units form the investigated department.

In the next step, the structure of detected clusters has been compared with the structure of organizational departments. Table 1 presents all detected clusters by HC algorithm with information about the number of vertices belonging to

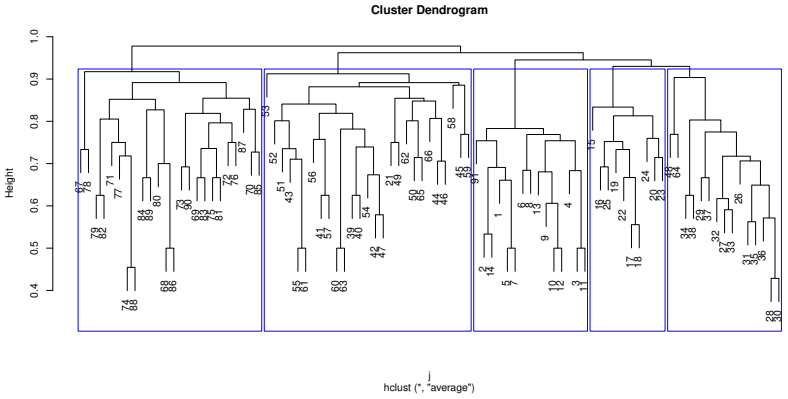


Fig. 1. Dendrogram showing clusters detected by HC algorithm.

them (in brackets) with a detailed list of these vertices. The number of vertices correctly assigned to clusters (when compare to the structure of the department) was 87 out of 91 (96%). In case of 4 out of 91 vertices (4%) they have been assigned incorrectly (presented in bold in Table 1 - 22, 49, 65, 68).

Table 1. Division of the network into communities by HC algorithm.

Clusters	Allocation of vertices to the communities
C1 (15)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
C2 (10)	16 17 18 19 20 21 23 24 25 26
C3 (15)	27 28 29 30 31 32 33 34 35 36 37 38 39 **49 65**
C4 (27)	**22** 40 41 42 43 44 45 46 47 48 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 66 67
C5 (24)	**68** 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91

In the last step of the analysis, results obtained by HC algorithm have been compared with the results obtained by Louvain method [2]. In one of the last Authors' paper [1], it has been shown that Louvain method can efficiently detect communities in studied social network. It turned out that division the network into communities obtained by both methods is similar. Moreover, the value of modularity index calculated for communities detected by both algorithms is almost the same (Louvain method - 0.504965, HC - 0.491054).

4 Conclusions

Hierarchical clustering method with dedicated similarity measure has been used to solve the problem of community detection in organizational social network based on e-mail communication. Computational experiment allowed one to discover communities which in most cases refer exactly to the structure of the organization. An interesting direction of the future work will be an investigation of different similarity measures between vertices as a fundamental part of different clustering methods dedicated to community detection problems.

References

1. Barbucha, D., Szyman, P.: Detecting Communities in Organizational Social Network Based on E-mail Communication. In: Czarnowski, I. et al. (eds.) Intelligent Decision Technologies. Smart Innovation, Systems and Technologies, vol 238, pp. 15–24. Springer, Singapore (2021).
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**, P10008 (2008).
3. Christidis, P.: Intensity of Bilateral Contacts in Social Network Analysis. Information **11**, 189 (2020).
4. Coscia, M., Giannotti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. Stat. Anal. Data Min. 4 (5), 512-546 (2011).
5. Fortunato, S., Hric, D.: Community detection in networks: A user guide. Phys. Rep. **659**, 1-44 (2016).

Forecasting the network traffic with PROPHET*

Paweł Cembaluk, Jakub Aniszewski, Aleksandra Knapińska[0000−0003−2654−4893], and Krzysztof Walkowiak[0000−0003−1686−3110]

Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wrocław, Poland
aleksandra.knapinska@pwr.edu.pl

Abstract. With the constant development of networking technologies and the increase of internet userbase, traffic prediction is becoming a vital part of today's network optimization. In this paper, we propose a method for network traffic prediction based on the PROPHET model. We examine its different parameters find their best configuration for diverse traffic types. Our research has shown that PROPHET is an accurate solution for backbone optical network traffic forecasting for a 14-day horizon.

Keywords: Traffic prediction · Application-aware network · Timeseries.

1 Introduction

The overall traffic in today's backbone optical networks is seeing tremendous growth in the last few years, especially during the COVID-19 pandemic. Various network-based services are widely used in many areas, including education, business, and entertainment. In turn, the summary traffic consists of multiple low-bitrate flows and thus is characterized by strong daily and weekly seasonality with periodically recurring patterns [3]. In light of the inevitable *capacity crunch*, various solutions are being developed for more efficient use of the existing network resources. Multilayer application-aware network optimization [6] is seen as a promising approach, in which different types of traffic related to various services and applications are treated according to their specific quality of service (QoS) requirements. The knowledge regarding future volumes of traffic used in network optimization algorithms can improve their efficiency and decrease bandwidth blocking [7].

In this article, we present a traffic prediction method based on PROPHET – a solution proposed by Facebook [5]. The PROPHET is a forecasting procedure for time data series based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. This method proved its effectiveness in the prediction of cellular network traffic [1,2,8] but, to the best of our knowledge, has not been studied in the context of backbone optical

* This work was supported by National Science Centre, Poland under Grant 2019/35/B/ST7/04272.

networks. To fill this research gap, we propose a throughout study of various parameters of PROPHET and evaluate it on datasets corresponding to diverse types of backbone optical network traffic. The rest of the paper is organized as follows. In Sec. 2, we define the problem and describe the chosen method. Sec. 3 describes conducted numerical experiments. Sec. 4 concludes this work.

2 Problem definition and method description

The problem considered in this work concerns the prediction of traffic in backbone optical networks. The created regression model attempts to predict future volumes of a specific traffic type with a given sampling rate based on historical data. The chosen forecast horizon is 14 days.

The PROPHET algorithm [5] provides automated timeseries forecasts that can be tuned. In this paper, we explore the impact of three of the PROPHET parameters on the prediction quality of various network traffic types. The first parameter is the changepoint prior scale, and it determines how much the trend changes at changepoints. Its value is a tradeoff between trend under- and overfitting. The second parameter is the seasonality prior scale. Its large value allows the seasonality to fit large variations, and a lower one decreases the magnitude of the seasonality. Because of the density of input data, as the last parameter, we chose the number of automatically placed changepoints.

As a reference approach, we propose a Linear Regression model, which proved to be an accurate method for network traffic forecasting [4]. Our implementation, adjusted for long-term traffic forecasting, takes four input features: the hour of the day, minute of the day, second of the day, and weekday.

3 Numerical experiments

The datasets in our experiments contain data based on the information from the Seattle Internet Exchange Point (SIX), collected between 22 X 2019 and 23 XII 2019, with a sampling rate of 5 minutes. To simulate diverse traffic types in a network, we added some fluctuations into the original data. To measure how the traffic in created datasets differs from the collected aggregated Seattle measurements, we use the mean absolute percentage error (MAPE). In this paper, we consider three datasets of diverse profiles: *traffic a* (MAPE = 3.39%), *traffic b* (MAPE = 8.21%) and *traffic c* (MAPE = 13.35%). Low MAPE values imply fewer fluctuations since the traffic is the most similar to the original aggregated SIX traffic. Intuitively, high MAPE values mean more fluctuations. The provided MAPE values are averaged across all the samples in each dataset. For more details regarding the datasets creation, we refer to [4].

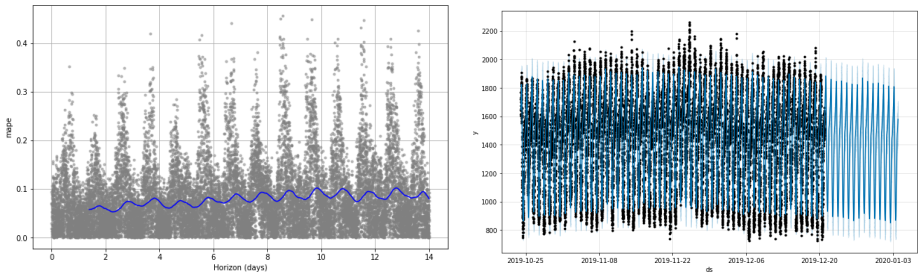
We explore the impact of PROPHET parameters on each traffic type, to obtain the most accurate prediction. For the changepoint prior scale, we investigate the values of 0.0005, 0.001, 0.01, 0.1, and 0.5. For the seasonality prior scale, the default value in the PROPHET is 10.0, which means almost no tuning. In our experiments, the tested values are 0.1, 1.0, and 10.0. For the number of

automatically placed changepoints, the default value in PROPHET is only 25, and in our experiments, the tested values are 25, 50, and 100. Overall, for each traffic type, we examine 45 parameter configurations.

Table 1: MAPE values for three best PROPHET configurations and the reference approach for each traffic type

	changepoint prior scale	seasonality prior scale	number of changepoints	MAPE
traffic a	0.001	1.0	25	0.07777
	0.001	0.1	100	0.07780
	0.001	10.0	25	0.07797
		linear regression		0.22504
traffic b	0.001	1.0	100	0.11837
	0.001	0.1	100	0.11845
	0.001	10.0	100	0.11874
		linear regression		0.23897
traffic c	0.001	1.0	25	0.15992
	0.001	10.0	25	0.16004
	0.0005	10.0	25	0.16007
		linear regression		0.25212

In Table 1, we present the results of the three best PROPHET configurations for each traffic type together with their parameters, alongside the reference Linear Regression model. Intuitively, the most accurate forecasts were obtained for the least fluctuating *traffic a*, and the highest prediction errors were noted for the most difficult *traffic c*. Performing the experiments, we noticed that the most significant PROPHET parameter is the changepoint prior scale. Its smallest value resulted in the lowest MAPE across traffic types. In particular, for *traffic b*, the difference between the worst result for the changepoint prior scale of 0.001 and the best for 0.5 was as high as four percentage points. After setting its value too low, i.e., 0.0001, the optimization algorithm failed – the model had to fall back to the Newton algorithm, yielding worse results than its default one. The seasonality prior scale did not seem to affect the results as much. In general, lower values, i.e., 0.1 and 1.0, yielded lower errors than 10.0. The differences between them were, however, only fractions of a percent. Furthermore, the number of changepoints did not appear to have any significant impact on the results either. Its influence was the most noticeable for cases with a changepoint prior scale of 0.01, where a higher number of changepoints resulted in more accurate forecasts. Once again, the results differ by only a fraction of percent, whereas more changepoints require more computational power for calculating the traffic prediction. Thus, it may not be worth it to increase this parameter. Overall, the predictions made by the PROPHET were of significantly higher quality than the reference Linear Regression. Though this approach appears in the literature as a prominent solution for short-term traffic forecasting (e.g., [4]), the PROPHET is versatile for long-term network traffic prediction.



(a) (b)

Fig. 1: PROPHET performance for *traffic a*. (a) Rolling window MAPE for best prediction model; (b) Prediction of the best model made for a 14 days horizon

An illustration of the results can be found in Fig. 1a and 1b where we present plots for the best model that we received during the experiments for *traffic a*. Fig. 1a shows MAPE with a 10% rolling window over the results. The grey points are the errors for each predicted point in the horizon. Figure 1b shows the input traffic data (in black) along with its predictions (in blue), and a 14-day forecast.

4 Conclusion

In this paper, we investigated the topic of optical backbone network traffic prediction. The developed model based on Facebook's PROPHET proved its high performance making 14-day forecasts for diverse types of network traffic. For each traffic type, various parameter configurations were tested to find the most accurate model. We found that the most important parameter in the PROPHET model for network traffic prediction is changepoint prior scale. In future work, we plan to use traffic forecasts for the optimization of application-aware networks.

References

1. Chakraborty, P., et al.: A comparative study for time series forecasting within software 5g networks. In: 14th Int. Conf. on Signal Proc. and Comm. Systems (2020)
2. Li, Y., et al.: Prophet model and gaussian process regression based user traffic prediction in wireless networks. Sc. China Information Sciences **63**(4) (2020)
3. NOKIA: The Nokia Deepfield Network Intelligence Report 2020. Tech. rep. (2020)
4. Szostak, D., et al.: Machine learning classification and regression approaches for optical network traffic prediction. Electronics **10**(13), 1578 (2021)
5. Taylor, S.J., et al.: Forecasting at scale. The American Statistician **72**(1), 37–45 (2018)
6. Tomkos, I., et al.: Application aware multilayer control and optimization of elastic wdm switched optical networks. In: 2018 OFC Conference (2018)
7. Troia, S., et al.: Deep learning-based traffic prediction for network optimization. In: 2018 ICTON Conference. IEEE (2018)
8. Xu, M., et al.: Hybrid holiday traffic predictions in cellular networks. In: NOMS 2018 IEEE/IFIP Network Operations and Management Symposium. IEEE (2018)

Multiple-Language Approach To Fake News Classification

Weronika Borek[0000−0003−2426−9541] and Paweł Ksieniewicz[0000−0001−9578−8395]

Department of Systems and Computer Networks,
Faculty of Information and Communication Technology,
Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{weronika.borek,pawel.ksieniewicz}@pwr.edu.pl

Abstract. In the era of the open Internet, which opened the door to global, communication, the risk of encountering false information has increased significantly as they are unverifiable. Fake news often becomes practically indistinguishable from ordinary news and their multitude forces the use of automation. This paper addresses this need by presenting an approach based on the use of multiple languages to diversify classifiers ensemble.

Keywords: Fake News · Classifier Ensemble · Natural Language Processing

1 Introduction

New amounts of data and information are produced every day around the world. The Internet has made it possible for every person to be not only a recipient, but also a sender of communicates that reach a wide range of people. This allowed for a significant improvement in the functioning of many branches of society and industry, but it also entailed an escalation of negative phenomena, such as fake news. Their influence on recipients' decisions was proved, among others, during the US elections in 2020, when Twitter and other social media were flooded with a wave of unconfirmed information about the voting procedure[1]. The current events related to the war in Ukraine, which from the beginning is surrounded by fake news and propaganda, also allow taking closer look at the dangers of disinformation.

The scale of the phenomenon prioritize the task of verifying information. It becomes easier to observe that handling this problem without far-reaching automation is impossible [5]. This paper proposes an approach to recognize fake news, which increases the quality of classification not by using complex *Natural Language Processing* algorithms or *deep neural networks*, which are already widely discussed in the field [2], but by data analysis in many languages simultaneously. In this way, the classifier ensemble is provided with an appropriate

[1] https://blog.twitter.com/en_us /topics/company/2020/2020-election-changes.html

diversification. Thanks to it, method can achieve better quality, according to the theory that a suitably diverse pool of even weak experts is capable of performing better than a highly qualified single expert [3]. Additionally, the process of translating texts using automated tool reduces it to a more unified form, often eliminating the use of synonyms and generating repetitions instead. Something that is a disadvantage in the context of linguistic hygiene can become an important advantage in the context of automatic text analysis.

2 Experiments

The experimental evaluation was performed using the *scikit-learn* [6] library. *Gaussian Naive Bayes* was chosen as the base classifier due to the simplicity and speed of its operation. As a metric to evaluate the results, the *balanced accuracy score* was used, selected due to its immunity to potential data imbalance.

The analysis used a fragment of the NELA-GT-2020 [4] dataset, which contains two thousand objects marked with an appropriate label, consisting of the title and content of the article. It has been translated from English into 34 other languages using the Google API [1]. The translation process itself eliminated some texts that turned out to be too complex for the API to handle. Ultimately, the experiments were carried out on 1677 objects, 200 of which were extracted using *Term Frequency - Inverse Document Frequency* (TF-IDF).

To ensure the reliability of the results, the 5×2 CV protocol was used, and for each language the training and test sets were separated using the same masks. This approach provided the possibility of induction on different datasets while maintaining a common bias. In practice, this means that exactly the same objects are compared every time, and translation is

Table 1. Achieved *balanced accuracy score* for every language model and multilingual ensemble

CC	LANGUAGE	TITLE	TEXT	ENSEMBLE
de	German	0.822	0.820	0.871
da	Danish	0.832	0.832	0.874
no	Norwegian	0.780	0.841	0.859
sv	Swedish	0.821	0.827	0.869
en	English	0.840	0.864	0.897
is	Icelandic	0.823	0.821	0.879
yi	Yiddish	0.754	0.808	0.812
bg	Bulgarian	0.740	0.809	0.830
mk	Macedonian	0.824	0.851	0.892
bs	Bosnian	0.817	0.791	0.872
hr	Croatian	0.749	0.808	0.845
sl	Slovenian	0.786	0.752	0.827
sr	Serbian	0.808	0.795	0.869
pl	Polish	0.779	0.810	0.844
cs	Czech	0.715	0.810	0.790
sk	Slovakian	0.786	0.768	0.830
ru	Russian	0.786	0.794	0.842
uk	Ukrainian	0.798	0.820	0.872
be	Belorussian	0.832	0.817	0.872
he	Hebrew	0.811	0.832	0.875
ar	Arabic	0.816	0.864	0.897
et	Estonian	0.813	0.780	0.857
fi	Finnish	0.820	0.794	0.858
hu	Hungarian	0.776	0.791	0.851
es	Estonian	0.826	0.843	0.882
fr	French	0.825	0.841	0.868
pt	Portuguese	0.811	0.836	0.871
it	Italian	0.826	0.848	0.883
ro	Romanian	0.805	0.834	0.873
lv	Latvian	0.820	0.779	0.868
lt	Lithuanian	0.818	0.791	0.863
zu	Zulu	0.814	0.792	0.872
eo	Esperanto	0.813	0.804	0.861
sq	Albanian	0.816	0.799	0.848
tr	Turkish	0.802	0.835	0.876
ens	Ensemble	0.835	0.895	0.866

an additional preprocessing procedure
prior to vectorization of documents.

Table 1 shows the *balanced accuracy score* results for all languages using each of the keys (TITLE and TEXT). Three types of classifier sets are also presented: (a) built for a single language, but consisting of all keys, (b) built for a single key, but using all languages, (c) aggregating all available sets and attributes.

In addition to analyzing the accuracy of the created models, it was decided to compare the predictions made by classifiers based on different languages. For this purpose, the *balanced accuracy score* metric was also used, in which, instead of indicating the correct labels, the predictions of the fitted models in different languages were cross-verified. The results for the key set are shown in Figure 1 in the form of a heatmap, where the lightest shade – white – means that the predictions were identical, and therefore the darker the color, the smaller the similarity. Languages within individual language groups were analyzed in detail, as it could be assumed that they would be the most similar to each other. According to this measure, the similarity does not fall below 80% for any language within the language group.

3 Conclusions And Future Works

The experiments presented in Section 2 showed that the same classification model used for many languages achieves results that differ in some cases by more than 10 percent. This proves the different level of complexity of individual translations, and thus, that some of the translations used in the experiment are characterized by a different degree of problem difficulty.

Thanks to this diversity, it is possible to build diversified classifier ensembles. Even if the pool composed of all tested sets does not achieve a significantly higher result than all its members, it can be concluded that it is possible to obtain higher accuracy. To do this, it would be worth carrying out an analysis based on the integration of classifiers by language groups, thanks to which they could reinforce each other more effectively. Alternatively, and also worth exploring, an integration would be possible using the *Mixture of Experts* approach.

It is also worth looking at the solutions using more complex classification algorithms than naive Bayes in subsequent works, which could result in obtaining higher-quality results and more extensive research.

Acknowledgement

This work was supported by the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology.

References

1. Bisong, E.: An overview of google cloud platform services. Building Machine Learning and Deep Learning Models on Google Cloud Platform pp. 7–10 (2019)

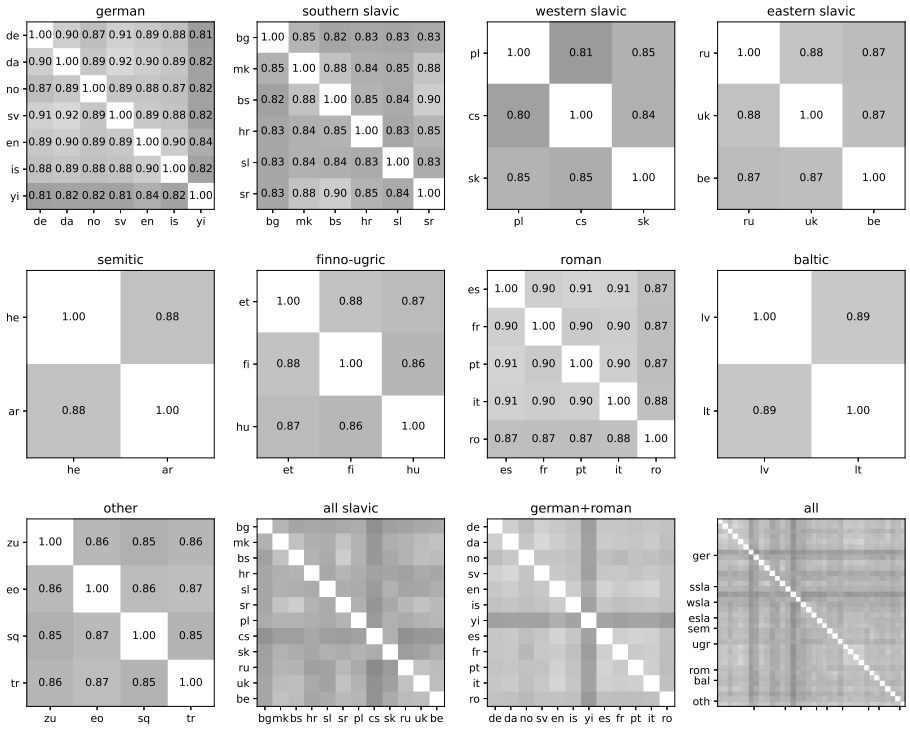


Fig. 1. Comparison of prediction cross compliance between languages from within different language groups according to balanced accuracy score

2. Chorąś, M., Demestichas, K., Giełczyk, A., Herrero, Á., Ksieniewicz, P., Remoundou, K., Urda, D., Woźniak, M.: Advanced machine learning techniques for fake news (online disinformation) detection: A systematic mapping study. Applied Soft Computing **101**, 107050 (2021)
3. Galton, F.: Vox populi (the wisdom of crowds). Nature **75**(7), 450–451 (1907)
4. Gruppi, M., Horne, B.D., Adalı, S.: Nela-gt-2020: A large multi-labelled news dataset for the study of misinformation in news articles. arXiv preprint arXiv:2102.04567 (2021)
5. Oshikawa, R., Qian, J., Wang, W.Y.: A survey on natural language processing for fake news detection. arXiv preprint arXiv:1811.00770 (2018)
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

The ATENA project: an international contribution to AI in Poland

Agata Giełczyk[1][0000−0002−5630−7461], Alfredo Petruolo[2], Michelangelo Agorini[2], Martyna Tarczewska[1], Karol Hartwig[1], and Gracjan Kątek[1]

[1] Bydgoszcz University of Science and Technology
Bydgoszcz, Poland
agata.gielczyk@pbs.edu.pl
[2] Università degli Studi di Napoli "Parthenope"
Napoli, Italy

Abstract. In this article we present the project ATENA: Artificial inTElligence traiNing progrAmme, which is an ongoing programme for students realized on the Bydgoszcz University of Science and Technology. During the programme foreign students have the opportunity to learn and get involved in the artificial intelligence. They follow some projects concerning the real-life issues and provide some promising results. The aim of this paper is to describe the ATENA project and its contribution to the AI development in Poland.

Keywords: ATENA · artificial intelligence · machine learning

1 Introduction

The Artificial Intelligence has become increasingly popular recently. The European Commission highlighted the significance of the AI development in Coordinated Plan on Artificial Intelligence which is also available online: *Artificial Intelligence (AI) can help us address some of the world's biggest challenges. It can enable doctors to improve diagnoses and develop therapies for diseases for which none exist yet; it can reduce energy consumption by optimising resources; it can contribute to a cleaner environment by lessening the need for pesticides; it can help improve weather prediction and anticipate disasters; and so on. The list is virtually endless. AI will be the main driver of economic and productivity growth and will contribute to the sustainability and viability of the industrial base in Europe. Like the steam engine or electricity in the past, AI is transforming the world*[1]. As the AI is one of the fastest developing element of computer science, there is still not enough AI specialists available on the market. This problem is especially visible in Poland, where the academia is not able to promote enough number of data science specialists and provide them to the industry.

2 The ATENA project

The ATENA (Artificial inTElligence traiNing progrAmme) project is the educational initiative held at Bydgoszcz University of Science and Technology in

Bydgoszcz, Poland as a part of the SPINAKER programme. Thanks to the support from National Agency for Academic Exchange it was possible to create the innovative, short and intensive course for foreign students. The main goal of each of three editions of the ATENA is to gather students, scientists and professionalists, and to provide them the space to share knowledge, discuss and create some powerful, realistic AI-based solutions. The key point of the project is to use numerous modern learning techniques. We worked with students in a blended form: there were both online and on site participants. Lectures were as well remote and in person. We used also the online resources bank provided for self-learning. The project management was performed mostly online due to the SARS-CoV-2 pandemic.

3 Student's projects

3.1 Predictions of biomedical properties based on low-coherence measurements

The main goal of this project was to predict if given sample is healthy or has cervical cancer. In this approach optoelectronic sensor was connected with machine learning algorithm to achieve this goal. Data was gathered by the optical measurements of refractive index of each sample as described in [4]. The data was pre-processed and enriched in following steps:

- Find local and global maximas,
- Filter data by threshold (0.05),
- Determine distance between local maximas,
- Get min/max wavelength,
- Amplitude of 'Amplitude' column,
- Calculate max/min/average/median of local maximas distances,
- Determine y value - if refractive index > 1.3 the tissue is infected otherwise is healthy,
- Get cavity length.

There were several models trained and evaluated: kNN, SVM, Decision Tree Classifier, Gaussian Naïve Bayes, Random Forest Classifier and XGBoost. The best performance on the dataset was achieved by XGBoost model. It provided the results reaching 98% accuracy, 98% precision, 100% recall and 99% F1.

3.2 Prediction the risk of defaulting a loan based on the customer behavior

The aim of the project was to implement a machine learning based methods to detect potential customers who may have problems with repaying a loan taken from a bank. In the research we used a publicly available dataset of people taking loans from banks provided by Kaggle. This collection consist of 280,000 rows and 13 columns, where each column, except last one, contains customer data.

The last column contains a label if customer paid back the loan. Before using machine learning algorithms, data had to be properly prepared. LabelEncoding was used for the column containing information about marital status and car ownership. On the other hand, OneHotEncoding was used for the city, state, house ownership and occupation columns. In order to implement the ML-based methods we divide the dataset as follows: 85% of the data was used for training and 15% for testing models. For the classification stage, three methods were used and compared: Decision Tree (DT), Random Forest (RF) and MLP classifier (MLP). Decision Tree and Random Forest achieved similar results at the score of $F1 = 77\%$ for DT and $F1 = 79\%$ for RF. The MLP provided the less promising results ($F1 = 53\%$).

3.3 Network intrusion detection

The aim of the project was to implement a machine learning-based methods in order to detect malicious activity on the Internet traffic data. In this research the dataset provided by UNB (University of New Brunswick's) was used. It is a collection of internet traffic in some days of the week. It is composed by 6 Million rows and 78 columns. Each column, apart the last one, represents a feature. The last one represents the label. In order to use the ML-based algorithms the pre-processing was needed: casting values (changing every value into numeric format) and handling NaN (the dataset was enough big to remove the records with NaN values). During the research 80% of samples were used for training and 20% for testing. For the classification step two methods were used and compared: Random Forest (RF) [2] and Artificial Neural Network (ANN) [3]. The ANN outperformed reaching the results reaching F1-score 88% - 99% for various classes, when the RF provided the F1-score reaching 67% - 93%.

3.4 Automation of document scanning

The aim of this project was the implementation of an automation mechanism for the recognition of documents. In this case the proposed method was used with sow cards from animal farms that have been previously filled manually and photographed. These documents need to be cropped and straightened, and later, significant data can be extracted from tables. The dataset provided by a local animal farms was very limited (only 17 photos taken in color with different illumination and angles conditions). Thus, the augmentation was needed: flipping, mirroring, brightness modification and shifting. The biggest challenge in the project was to perform the segmentation. Firstly, the masks were painted manually for the whole small 17-elements dataset. Then, the U-net architecture, which is considered as very powerful in such applications [5], was used with satisfying results. It was possible to obtain the accuracy reaching 95%.

4 Conclusions

As presented above, the AI methods can be implemented for solving numerous problems. The propositions of the students' solutions were created during an one week long stay in Poland. Thus, in the future they can be improved and investigated in details. However, the project was an opportunity of students to cooperate in the international teams, to discover some new AI-based architectures and to make the first approach in designing the scientific experiment and run the research.

The AI is an extremely important issue in the computer science studies, but it is often underappreciated in the higher studies programs. ATENA project shows that data science and machine learning can be very interesting and involving for students.

Acknowledgements

This project has received founding from the European Union's PO WER project. It is supported by the Polish National Agency for Academic Exchange in the SPINAKER programme. The total EU funding is 369 490,00 PLN.

References

1. Coordinated plan on artificial intelligence (2018), https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52018DC0795&rid=3
2. Ajdani, M., Ghaffary, H.: Introduced a new method for enhancement of intrusion detection with random forest and pso algorithm. Security and Privacy **4**(2), e147 (2021)
3. Choraś, M., Pawlicki, M.: Intrusion detection approach based on optimised artificial neural network. Neurocomputing **452**, 705–715 (2021)
4. Kruczkowski, M., Drabik-Kruczkowska, A., Marciniak, A., Tarczewska, M., Kosowska, M., Szczerska, M.: Machine learning for predictions of cervical cancer identification–preliminary investigation based on refractive index. preprint available at ResearchSquare (2021)
5. Siddique, N., Paheding, S., Elkin, C.P., Devabhaktuni, V.: U-net and its variants for medical image segmentation: A review of theory and applications. IEEE Access **9**, 82031–82057 (2021)

WearApp – Data Gathering in Emotion and Affect Experiments with Wearables

Mateusz Ficek and Paweł Jemioło
AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
mficek@student.agh.edu.pl, pawljmlo@agh.edu.pl

Abstract. Wearable devices have become increasingly popular due to low prices. Physiological data gathered using smartwatches and smartbands can be used for emotion and affect recognition with Artificial Intelligence (AI). Still, with a variety of available devices, there is no standard mobile software that enables researchers to experiment with them. In this paper, we present WearApp – a Flutter application that allows for simultaneous data acquisition from popular Bluetooth Low Energy and Bluetooth wearables. We validate the presented software in several scenarios, thus confirming it can be applied in different laboratory settings, including gathering data for emotion and affect recognition experiments. WearApp is an open-source solution for Android and iOS.

Keywords: Emotion and affect recognition · Data gathering · Wearables · Flutter · Application · Mobile · Android · iOS

1 Introduction

Wearable devices have become increasingly popular due to relatively low prices and growing availability. Sensors initially developed for specific purposes, i.e. healthcare, robotics or military, find their way into mainstream gadgets accessible to everyone. On a daily basis, humans can obtain data about their movement, heart rate, glucose level, electrodermal activity, and many others [5].

In the beginning, such devices were designed to inform users about basic statistics, e.g. number of steps made and calories burnt during the day. With progressing miniaturisation, more advanced sensors were introduced into wearables. For example, they allow for developing software able to measure idle time or determine sports activity with no direct human control.

Moreover, wearable devices are frequently used in emotion and affect recognition trials [5]. They are small and comfortable for participants; therefore, applying them in research may increase the ecological validity of scientific experiments reducing the *garbage in, garbage out* problem.

Enabling emotion and affect recognition is crucial to enhancing people's everyday life. There are numerous scenarios where it can be implemented, e.g. smart homes, personalised medicine and games that adapt to one's feelings.

In general, due to its complexity, emotion and affect recognition require using different physiological signals at the same time [5]. Thus, the researchers often apply more than one device for measuring various modalities. Obtaining required signals may be laborious, with separate applications designed for a specific wearable. Additionally, some of them do not provide access to raw data.

In this paper, we aim to present an open-source Flutter application for iOS and Android that may be used for flexible data gathering in experiments with multiple wearable devices simultaneously.

2 Background

When manufacturing wearables, some limitations need to be addressed. First, the modest size of such devices forces implementing energy-efficient solutions. The use of communication protocols, e.g. NFC or Bluetooth, is necessary to ensure constant data exchange between wearable and main device (in most cases a smartphone) and eliminate the need for high-capacity memory [3].

Modern wearables most often implement Bluetooth Low Energy (BLE) standard. Its major limitation is the data size that can be transferred in one moment. However, this constraint does not affect wearables since the data they send is on average 30 bytes [2]. The older protocol used in wearable gadgets is a classic Bluetooth. It uses considerably more battery when compared to BLE [6].

Data exchange differs in presented technologies. Most devices follow the universal Generic ATTribute Profile (GATT) standard [6]. It defines how two BLE devices exchange data. However, some manufacturers tend to use custom made Bluetooth communication. These custom communication need to be handled individually based on producers' specifications. On the other hand, GATT allows for creating a unified approach for a broad range of devices.

Communication with wearable devices requires a Bluetooth module, universally used in smartphones. Current smartphones operate with iOS or Android operating systems. Both of those platforms have their advantages, but they require separately built applications. It causes an increase in resources and development time. Rather than creating applications with platform-intended languages, developers may use multi-platform frameworks. Flutter is one of them and supports writing applications for Android, iOS, and the web.

As the wearables are frequently used for gathering data in emotion and affect recognition experiments, the researchers incorporate them in different ways – using native manufacturers' applications and custom made ones. See the paper by Alpers and Benta [1] for a comprehensive overview. However, there is still no standard open-source application for BLE and Bluetooth wearable devices that operates both on iOS and Android smartphones.

3 Presented approach

Based on our previous experiments within emotion and affect recognition [4], we pre-specified requirements for the presented application: adding support for the new device should be facile; it needs to operate on iOS and Android; both classic Bluetooth and BLE devices need to be supported; raw data from a specific device should be recorded and saved in CSV file immediately after receiving it.

WearApp was written in Flutter with support for Android and iOS. The application consists of 3 main screens presented in Figure 1. First screen (starting from left) represents available devices with highlighted connected gadgets. The second and third screens present BLE and Bluetooth data gathering views. The fourth one sums up the number of gathered samples.

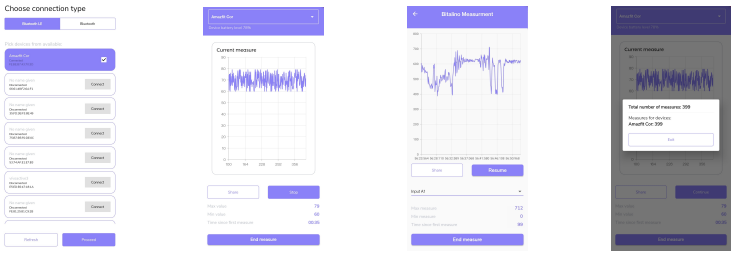


Fig. 1: Four screenshots of WearApp presenting device selection, data collection (BLE and classic Bluetooth) and measurement summary, from left respectively.

For BLE devices, WearApp, for now, supports heart rate values but can be easily expanded to handle more sensors. GATT services and characteristics work as a lookup table, so acquiring different data requires only reading information with a different ID. That can be achieved by extending `BLEManager` class.

After entering the app, the user is presented with a list of available devices. Pressing on one of them establish the connection with the selected equipment. Then, the user can proceed to the data gathering screen (*Proceed* button). Pressing *Resume* or *Continue* or *Stop* buttons toggle between working and paused states. *End measure* finishes acquisition from wearables.

As for BLE accessories, the user can connect to multiple devices at once. In the visualisation area, information about the selected accessory is presented. The data include battery level, last measured heart rate, a chart with the last 300 measures, maximum and minimum heart rate, and measurement duration. There is also a drop-down button that enables switching between accessories.

As for Bluetooth equipment, the software currently supports Bitalino device. WearApp can be easily expanded to handle more devices, as it only requires implementing connection protocol with new gadgets in the `BluetoothDevices` class and calling existing functions that visualise and save acquired data. Adding support for more BLE sensors in code is not mandated as it works out of the box thanks to implemented standards. As for visualisation, the user can switch between possible input charts for every simultaneously acquired input signal.

Collected data is instantly saved to a CSV file. After creating a new file in smartphone storage, each sample received from the device is converted and added at the end of a file. Every entry is converted to a row, including measurement ID, gathered data, and sample date. After the acquisition, the created document can be saved with a custom name, e.g. person ID and shared.

The code of this application, as well as Android installation files, are available on GitHub: https://github.com/mateuszFicek/wearapp.

4 Results

We tested the presented application in several different scenarios involving: acquiring connection between devices, connection stability, and reading and quality of gathered data.

The experiments were performed using the OnePlus 6T smartphone with wearable devices using a classic Bluetooth (Bitalino (r)evolution kit) and BLE technology (Polar H10, Scosche Rhythm+, and Xiaomi MiBand 5).

Based on 10 attempts for each wearable, only Bitalino had problems with the connection. Still, we achieved an 80% success rate in terms of this device and 100% for others. To verify the stability, each device gathered data for 300 to 1500 seconds, then the trial was paused, the screen was locked, and the acquisition was resumed. The test was passed for each device with pauses from 300 to 1,200 seconds. The quality of measurements was tested by acquiring data for 60, 300, 6,000, 12,000 and 18,000 seconds. Next, we compared the number of samples and maximum and minimum values between WearApp and CSV file. The collected data did not differ from those presented in the application in any case.

5 Conclusions and Future Work

In this paper, we present WearApp – an open-source Flutter application for iOS and Android that may be used for flexible data gathering in experiments with wearable devices. We tested the software with scenarios similar to real-world emotion and affect recognition experiments. Data collected using WearApp may be used in developing AI models in this area with supervised learning techniques.

In the future, we want to support more signals than cardiovascular ones in BLE apparatuses. Further improvements to the application could include automatic measurement management and voice control. Apart from that, we want to implement live data processing and include basic emotion and affect recognition algorithms directly in WearApp.

Funding Sources

Research project supported by program *Excellence initiative -– research university* for the University of Science and Technology.

References

1. Alpers, S., Benţa, K.I.: Sensegraph: Affect self-monitoring and tagging tool with wearable devices. In: 2021 International Conference on Computer Communications and Networks (ICCCN). pp. 1–6. IEEE (2021)
2. Carles Gomez, Joaquim Oller, J.P.: Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. Sensors (2012)
3. Dragomir, D., Gheorghe, L., Costea, S., Radovici, A.: A survey on secure communication protocols for iot systems. In: 2016 International Workshop on Secure Internet of Things (SIoT). pp. 47–62. IEEE (2016)
4. Kutt, K., Nalepa, G.J., Giżycka, B., Jemiolo, P., Adamczyk, M.: Bandreader-a mobile application for data acquisition from wearable devices in affective computing experiments. In: 2018 11th International Conference on Human System Interaction (HSI). pp. 42–48. IEEE (2018)
5. Saganowski, S., Dutkowiak, A., Dziadek, A., et al.: Emotion recognition using wearables: A systematic literature review-work-in-progress. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp. 1–6. IEEE (2020)
6. Townsend, K., Cufí, C., Davidson, R., et al.: Getting started with Bluetooth low energy: tools and techniques for low-power networking. O'Reilly Media, Inc. (2014)

Author index