Zbornik strokovne konference

ROSUS 2020

Računalniška obdelava slik in njena uporaba v Sloveniji 2020

Urednik:
Božidar Potočnik

Maribor, 19. marec 2020

Univerzitetna založba
Univerze v Mariboru

LSPO
Laboratorij za sistemsko
programsko opremo

Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Inštitut za
računalništvo

ROSUS 2020 -
Računalniška obdelava slik in njena uporaba v Sloveniji 2020

Zbornik 15. strokovne konference

Urednik
Božidar Potočnik

Marec 2020

Naslov *Title*
ROSUS 2020 - Računalniška obdelava slik in njena uporaba v Sloveniji 2020
ROSUS 2020: Computer image processing and its application in Slovenia 2020

Podnaslov *Subtitle*
Zbornik 15. strokovne konference
Proceedings of the 15th Professional Conference

Urednik *Editor*
Božidar Potočnik
(Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)

Recenzija *Review*
Aleš Holobar
(Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)

Simon Dobrišek
(Univerza v Ljubljani, Fakulteta za elektrotehniko)

Tehnični urednik *Technical editor*
Jan Perša
(Univerza v Mariboru, Univerzitetna založba)

Oblikovanje ovitka *Cover designer*
Gašper Sedej
(Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)

Grafične priloge *Graphic material*
Avtorji prispevkov

Konferenca *Conference*
Strokovna konferenca ROSUS 2020:
Računalniška obdelava slik in njena uporaba v Sloveniji 2020

Datum konference *Conference date*
19. marec 2020

Datum konference *Conference location*
Maribor, Slovenija

Založnik / *Publihed by*
Univerza v Mariboru
Univerzitetna založba
Slomškov trg 15, 2000 Maribor, Slovenija
https://press.um.si, zalozba@um.si

Izdajatelj / *Co-published by*
Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Koroška cesta 46, 2000 Maribor, Slovenija
https://feri.um.si, feri@um.si

Izdaja *Edition*
Prva izdaja

Vrsta publikacije *Publication type*
E-knjiga

Dostopno na *Available at*
https://press.um.si/index.php/ump/catalog/book/464
https://rosus.feri.um.si/rosus2020

Izdano *Published*
Maribor, marec 2020

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

004.932(497.4)(082)(0.034.2)

STROKOVNA konferenca ROSUS (15 ; 2020 ; Maribor)
ROSUS 2020 [Elektronski vir] : računalniška obdelava slik in njena uporaba v Sloveniji 2020 : zbornik 15. strokovne konference / urednik Božidar Potočnik. - 1. izd. - Maribor : Univerzitetna založba Univerze, 2020

Način dostopa (URL): https://press.um.si/index.php/ump/catalog/book/464
ISBN 978-961-286-337-1 (pdf)
doi: 10.18690/978-961-286-337-1
1. Potočnik, Božidar
COBISS.SI-ID 98306561

ISBN	978-961-286-337-1 (pdf) 978-961-286-338-8 (Mehka vezava)
DOI	https://doi.org/10.18690/978-961-286-337-1
Cena *Price*	Brezplačni izvod
Odgovorna oseba založnika *For publisher*	prof. dr. Zdravko Kačič, rektor Univerze v Mariboru

ROSUS 2020 - RAČUNALNIŠKA OBDELAVA SLIK IN NJENA UPORABA V SLOVENIJI 2020

BOŽIDAR POTOČNIK

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Maribor, Slovenija, e-pošta: bozidar.potocnik@um.si

Povzetek ROSUS 2020 – Računalniška obdelava slik in njena uporaba v Sloveniji 2020 je strokovna računalniška konferenca, ki jo od leta 2006 naprej vsako leto organizira Inštitut za računalništvo iz Fakultete za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru. Konferenca povezuje strokovnjake in raziskovalce s področij digitalne obdelave slik in strojnega vida z uporabniki tega znanja, pri čemer uporabniki prihajajo iz raznovrstnih industrijskih okolij, biomedicine, športa, zabavništva in sorodnih področij. Zbornik konference ROSUS 2020 združuje strokovne prispevke več deset avtorjev, od tega dva vabljena predavanje ter več demonstracijskih prispevkov. Prispevki podajajo najnovejše dosežke slovenskih strokovnjakov s področij digitalne obdelave slik in strojnega vida, osvetljujejo pa tudi trende in novosti na omenjenih strokovnih področjih. Velik poudarek prispevkov je na promoviranju ekonomske koristnosti aplikacij računalniške obdelave slik in vida v slovenskem prostoru. Takšne računalniške aplikacije zaradi visoke natančnosti, robustnosti in izjemnih hitrosti pri obdelovanju informacij nudijo namreč nove priložnosti za uveljavitev na trgu visokih tehnologij.

Ključne besede: računalniška obdelava slik, strojni vid, biomedicina, industrijske aplikacije, prenos znanja.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1
ISBN 978-961-286-337-1

ROSUS 2020 - COMPUTER IMAGE PROCESSING AND ITS APPLICATION IN SLOVENIA 2020

BOŽIDAR POTOČNIK
University of Maribor, Faculty of Computer Science and Informatics, Maribor Slovenia,
e-mail: bozidar.potocnik@um.si

Abstract ROSUS 2020–Computer image processing and its application in Slovenia 2020 is a professional conference that, since 2006, has been organised every year by the Institute of Computer Science of the Faculty of Electrical Engineering and Computer Science, University of Maribor. The conference connects researchers in the fields of Image Processing and Machine Vision with users of this knowledge, whereby users are coming from diverse industrial environments, such as Biomedicine, Sport, Entertainment, and related fields. The proceedings of ROSUS 2020 combine scientific articles by dozens of authors, including two invited lectures and several demonstration articles. Contributions represent the latest achievements of Slovenian experts in the fields of Image Processing and Vision, and also highlight trends and novelties in these areas. Great emphasis is on promotion of the economic usefulness of Image Processing and Vision applications in the Slovenian region. Namely, such software, due to high precision, robustness, and exceptional speed in information processing, provides new opportunities for penetration on the high technologies market.

Keywords: computer image processing, machine vision, biomedicine, industrial applications, knowledge transfer.

https://doi.org/10.18690/978-961-286-337-1 DOI
978-961-286-338-8 ISBN

University of Maribor Press

Univerzitetna založba
Univerze v Mariboru

Kazalo

Spoštovani!

Po štirinajstih konferencah ROSUS 2006–2019 želimo tudi s konferenco ROSUS 2020 nadaljevati s promoviranjem pomembnosti ekonomske koristi računalniške obdelave slik na področjih industrije, biomedicine in drugih poslovnih procesov. Vezi, ki smo jih na prejšnjih konferencah stkali med raziskovalci, razvijalci, ponudniki rešitev ter uporabniki računalniške obdelave slik v slovenskem prostoru, želimo še dodatno okrepiti, ob tem pa nuditi tudi možnosti sklepanja novih sodelovanj in svetovanja pri razreševanju konkretnih poslovnih oziroma raziskovalnih problemov.

Tudi letos namenjamo glavni poudarek aplikacijam s področja računalniške obdelave slik, ki so že integrirane oziroma pripravljene za integracijo v poslovne procese. Na tej konferenci nadaljujemo globalni trend na področju računalniškega vida s popoldansko sekcijo z naslovom »Globoko učenje: Praktični nasveti strokovnjakov«, ki smo jo organizirali v sodelovanju s podjetjem Kolektor. Demonstrirali bomo, da avtomatska obdelava v industriji lahko zaradi svoje natančnosti in hitrosti prinaša velike ekonomske koristi, hkrati pa nakazali, da aplikacije računalniške obdelave slik nudijo nove priložnosti za uveljavitev na trgu visokih tehnologij. Seveda ne smemo pozabiti na možnost diskusije ter predstavitev konkretnih problemov in potreb, ki se porajajo pri uporabnikih, s katerimi bomo računalniško obdelavo slik in njeno koristnost še bolj približali avditoriju.

Naj sklenemo uvodne misli še s prisrčno zahvalo Javnemu skladu Republike Slovenije za podjetništvo, ki je v okviru konference ROSUS 2020 predstavil zanimive finančne instrumente za spodbujanje prenosa tehnoloških rešitev v podjetniško sfero. Izpostaviti želimo še medijskega pokrovitelja revijo IRT3000, ki je intenzivno promoviral konferenco ROSUS 2020 ter pomen strojnega vida v slovenskem prostoru.

Božidar Potočnik
predsednik konference
ROSUS 2020

Programski odbor ROSUS 2020:

dr. Aleš Holobar
dr. Damjan Zazula
dr. Boris Cigale
dr. Marjan Mernik
dr. Peter Peer
dr. Tomaž Tomažič
dr. Matjaž Colnarič
dr. Slobodan Ribarić
dr. Vojko Flis
dr. Saša Divjak

dr. Stanislav Kovačič
dr. Zdravko Kačič
dr. Aleš Leonardis
dr. Simon Dobrišek
dr. Mitja Lenič
dr. Franc Solina
dr. Andrej Šoštarič
dr. Franjo Pernuš
dr. Jure Skvarč
dr. Božidar Potočnik

Organizacijski odbor 2020:

Gašper Sedej
Jurij Munda
dr. Uroš Mlakar
dr. Matjaž Divjak
Martin Šavc
Jernej Kranjec
dr. Danilo Korže
dr. Božidar Potočnik

POKROVITELJI

IEEE
Slovenska sekcija

SDRV
Slovensko društvo za razpoznavanje vzorcev
Slovenian Pattern Recognition Society

SLOVENSKI PODJETNIŠKI SKLAD

ROSUS 2020
http://rosus.feri.um.si

VABLJENA PREDAVANJA

SLIKOVNA BIOMETRIJA NA POHODU

PETER PEER

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija,
e-pošta: peter.peer@fri.uni-lj.si

Povzetek V zadnjih petih letih se je v Laboratoriju za računalniški vid na FRI UL oblikovala močna skupina, ki dela na področju biometrije. Prvi ključni koraki so bili narejeni v okviru kompetenčnih centrov, kjer smo v oblaku naredili fuzijo dveh modalnosti, obrazov in prstih odtisov. Vzporedno s tem se je odvijalo takrat tudi delo na razpoznavanju ljudi iz načina gibanja. Nato pa je delo na področju biometrije dobilo še dodaten zagon, posvetili smo se povsem novi modalnosti uhljev, začeli delati na izzivu fotorealistične deidentifikacije, dodali beločnico, šarenico ter obočesno regijo kot naslednje tri sveže modalnosti. Na drugi stopnji študija smo uvedli tudi nov izbirni predmet Slikovna biometrija. Ta ima letos kar 80 slušateljev. Število članov skupine trenutno raste iz leta v leto, temu primerno tudi publikacije na ključnih konferencah ter v revijah, nenazadnje pa se vpliv skupine pozna tudi pri organizaciji tekmovanj na teh ključnih konferencah ter tudi zmagah na sorodnih tekmovanjih. Predavanje bo osvetlilo prehojeno pot skozi ključne raziskovalne vsebine.

Ključne besede:
računalniški vid,
biometrija,
globoko učenje,
fuzija modalnosti,
deidentifikacija.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1
ISBN 978-961-286-337-1

POPOLDANSKA SEKCIJA

Globoko učenje: Praktični nasveti strokovnjakov

TOWARDS VISUAL ANOMALY DETECTION IN DOMAINS WITH LIMITED AMOUNT OF LABELED DATA

DEJAN ŠTEPEC & DANIJEL SKOČAJ

XLAB Research, XLAB d.o.o., Ljubljana, Slovenia, e-mail: dejan.stepec@xlab.si
University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, e-mail: danijel.skocaj@fri.uni-lj.si

Abstract Anomaly detection in visual data refers to the problem of differentiating abnormal appearances from normal cases. Supervised approaches have been successfully applied to different domains, but require abundance of labeled data. Due to the nature of how anomalies occur and their underlying generating processes, it is hard to characterize and label them. Recent advances in deep generative based models have sparked interest towards applying such methods for unsupervised anomaly detection and have shown promising results in medical and industrial inspection domains.

Keywords:
anomaly detection, unsupervised, deep-learning, autoencoders, generative adversarial networks.

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-337-1.1
ISBN 978-961-286-337-1

1 Introduction

Anomaly detection represents an important process of determining instances that stand out from the rest of the data. Detecting such occurrences in different data modalities is widely applicable in different domains such as fraud detection, cyber-intrusion, industrial inspection and medical imaging [1]. Detecting anomalies in high-dimensional data (e.g. images) is a particularly challenging problem that has recently seen a particular rise of interest, due to prevalence of deep-learning based methods.

Success of current deep-learning based methods has mostly relied on abundance of available data. Anomalies generally occur rarely, in different shapes and forms and are thus extremely hard or even impossible to label. Supervised deep-learning approaches have seen great success in different domains, including in anomaly detection [2]-[4]. Success of such methods is the most evident in the domains with well-known characterization of the anomalies and abundance of labeled data. Specific to the visual anomaly detection domain, we usually also want to localize the actual anomalous region in the image. Obtaining such detailed labels to learn supervised models is a costly process and in many cases also impossible. Weakly-supervised approaches address such problems by requiring only image-level labels and are thus able to infer anomalous regions solely from weakly labeled data [5]-[7]. In an unsupervised setting, only normal samples are available, which are usually available in abundance. Such methods represent the most general case and are the most widely applicable. Deep generative methods have been recently applied to the problem of unsupervised anomaly detection (UAD) and have shown promising results [8], [9]. Current methods are usually developed for a particular domain or on synthetic datasets which limits their generality, as well applicability to real-world applications. They are also not really unsupervised, requiring only normal samples, with significant drops in performance with the presence of small amount of contaminated training data [10], [11].

In this work we focus on anomaly detection from images, which was just briefly mentioned in one of the most significant papers on general anomaly detection [1]. This clearly shows the state of this domain before the era of deep-learning. There have been a lot of advancements in recent years in the visual anomaly detection domain, but there is no survey work that clearly summarizes them. Most of the existing survey papers are addressing the wider scope of anomaly detection problem, lacking the focus on visual anomaly detection and its recent advancements [1]. Some survey papers are addressing recently popular deep-learning based anomaly detection approaches [2], but are describing applications to the broader field of anomaly detection and are not focusing on particular methods and application domains related to visual anomaly detection. Similarly, there are survey papers that are focusing on a particular set of methods [12]. Our work is addressing some of this limitations, by providing a general overview and at the same time limiting the focus to a few application domains and representative state-of-the-art methods. We also explore and present open research problems, from methodological point of view, as well as novel challenging application domains, untapped by existing UAD methods.

2 Taxonomy of Learning Approaches

2.1 General Anomaly Detection

The general problem of anomaly detection, as well as domain specific applications have been a topic of a number of surveys and review articles [1], [2], [12]. In this work we emphasize survey paper [1], which provides an extensive overview, spanning multiple research areas and application domains. This survey paper is particularly interesting as it captures all the relevant research and application domains before the era of deep-learning and clearly shows the state of research interest towards visual anomaly detection.

Anomaly detection refers to the problem of differentiating abnormal appearances from normal cases. These abnormal appearances are in the literature known as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants, depending on the application domain [1]. Applications of anomaly detection can be found in fraud detection systems for credit cards and insurances, intrusion detection systems for cyber-security, industrial inspection and medical imaging. According to [1], anomalies can be categorized as point anomalies, contextual anomalies and collective anomalies. Point anomaly represents an individual data instance, that deviates from the rest the data and represents the simplest type of anomaly and is also the focus of research on anomaly detection. Point anomaly can be represented as a contextual anomaly, if it is not conforming to the expected behaviour in a specific context (e.g. a low temperature in summer).

Collective anomalies, on the other hand, represent a set of data points, which together represent a deviation from a normal behaviour.

Anomaly detection methods can also be used for novelty detection, as they offer capabilities to detect unseen patterns in data, that could translate to new actionable insights. The difference is that the novel patterns are usually used alongside the previously known patterns, after being detected.

2.2 Availability of Labeled Data

Availability of large scale datasets [13] with labeled data and proliferation of deep-learning based methods has brought tremendous improvements particularly to computer vision domain [14]. Obtaining labeled data is often very expensive, as it is usually done manually by a human expert. Obtaining labeled data for anomaly detection is even harder, or even impossible, due to the nature of anomaly occurrences and unknown underlying processes, that generate them. Important factor is also the level of details, that are provided with the labels. This is especially important for visual anomaly detection, where labels can be on the image level (i.e. contains anomaly or not) or at the pixel level, delineating location and the extent of anomalies.

Anomaly detection methods are categorized to the bellow presented modes, based on the extent, to which the labels are available [1].

1) Supervised anomaly detection: Methods trained in a supervised fashion require labeled data for normal, as well as anomalous cases. There is usually much more labeled data for normal instances, which makes this an extremely imbalanced classification problem. Generalization performance of such methods is usually worse, due to limited availability and fixed vocabulary of representative labels and can also vary by the application domain.

2) Semi-supervised anomaly detection: According to [1], semi-supervised anomaly detection methods require labeled data only for normal instances and are as such more widely applicable. Such classification is often interchangeably also used for current unsupervised anomaly detection approaches, where normal instances are usually implicitly labeled as normal. A more common and widely used semi-supervised setting is when there is a combination of large set of unlabeled samples and a small pool of labeled ones [15].

3) Weakly-supervised anomaly detection: Weakly-supervised anomaly detection has not been considered in [1], mostly due to recent advancements and applications of such methods in the domain of visual anomaly detection [5]-[7]. In the context of industrial inspection or anomaly detection in medical imagery, we want to detect the anomaly, as well as localize it. Detailed ground-truth localized annotations are expensive or impossible to obtain in many cases. Weakly-supervised anomaly detection approaches utilize only image-level labels (i.e. contains anomaly or not) and are able to localize anomalous regions, without pixel-level annotations in the case of visual anomaly detection.

4) Unsupervised anomaly detection: Unsupervised methods do not require any labeled data and are as such the most widely applicable. They are usually trained with normal samples only in order to learn the distribution and are later on capable of capturing out-of-distribution samples. These methods run on the assumption that normal samples are far more frequent than anomalous ones. With recently presented deep generative based methods, accurate detection and localization of anomalous regions is possible, without any supervision [8], [16]. In the literature most of these methods are treated as unsupervised, despite weak implicit supervision, which is introduced by selecting only normal samples for training. In real-life scenarios one should expect that there will be some small percentage of contaminated data in training samples [10], [11].

3 Visual Anomaly Detection

Visual anomaly detection is dealing with detecting and localizing anomalous regions in imagery data. We have seen great success in computer vision domain since the introduction of deep-learning based methods and consequently, visual anomaly detection has also seen increasing interest and success [2]. The primary benefit of deep-learning based methods is the data driven approach, which eliminates the need for the expert-level feature engineering, which has shown sub-optimal performance [14], [17]. Despite the proliferation of deep-learning based methods, there is relatively small amount of methods that are truly addressing anomaly detection problem, especially in a real-world setting.

In the next sections we discuss these recent improvements in the context of industrial inspection and medical domain. We first briefly present a few application domains and associated data, with the focus on two recently presented large-scale datasets. Later on we categorize the methods based on the availability of data and present the main representatives.

3.1 Datasets

Large-scale labeled datasets have been one of the main contributing factors to the recent success of deep-learning based methods [13], [14]. Due to the nature and frequency of anomaly occurrence in real-world application domains, it is difficult to obtain such large-scale datasets for anomaly detection.

Most of the initial work on visual anomaly detection has been performed on existing classification datasets [13], [18], [19], by considering a subset of the existing classes as anomalous samples and the rest of them as normal. With this approach one gets access to large-scale datasets to develop anomaly detection methods, but the anomalous samples differ significantly from the normal ones and are as such not representing real-world conditions. Manufacturing defects in industrial inspection domain or lesions in medical imagery are usually hard to detect and do not alter the resultant image, to differ significantly from normal samples. Equally important in visual anomaly detection domain is also the ability to segment anomalous regions, which is especially vital for industrial inspection and medical domain.

Real-world datasets for anomaly detection are rare, due to difficulties to create them, as well as due to confidential and privacy concerns. Industrial inspection is performed with industrial grade cameras [4] and specialized devices, such as X-Ray CT scans [20], which can reveal the details of the manufacturing process. Similarly, medical imagery can contain personal information and needs to be reviewed by medical boards and in some cases, patients' consents are needed [3]. Despite confidential and privacy concerns, there are some datasets, that have been made publicly available and two representative datasets, that will be used in our research work are presented next.

1) MVTec AD - A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection: The dataset presented in [21] represents the first comprehensive, multi-object, multi-defect dataset for anomaly detection in a real world scenario of an industrial inspection. In comparison with other works [22]-[24], that evaluate anomaly detection methods on existing classification datasets, this represents a much more realistic scenario, with anomalies manifesting in less significant differences from the training data. MVTec Anomaly Detection dataset consists out of 15 categories of different objects and textures. 3629 images are provided for training and validation and another 1725 for testing. The training data does not contain defects. Altogether, 73 different types of defects are encountered, with provided pixel-wise annotations for all the test examples. Example data, together with anomalies and pixel-wise annotations are presented in Figure 1. The captured dataset represents close-to real-world conditions, with some of the objects being rigid, while others deformable or with natural variations. Some of the objects are captured in aligned poses and some in random rotations. All images were acquired using 2048 x 2048 industrial grade RGB camera and resultant images were cropped to different resolutions between 700 x 700 and 1024 x 1024 pixels. Some of the images were intentionally provided in gray-scale and under different (uncontrolled) illumination conditions, to increase variability.

A thorough evaluation of multiple state-of-the-art unsupervised anomaly detection methods was performed. AnoGAN [16] method, based on generative adversarial networks (GANs), as well as a method based on auto-encoders and structural similarity [25] were evaluated. Both of the methods are described in section 4.3. Additionally, they evaluate a classical Convolutional Neural Network (CNN) feature extraction approach [26], as well as traditional non deep-learning methods based on Gaussian Mixture Model (GMM) [27] and a simple variational model approach [28].

Results were reported based on the classification, as well as anomaly segmentation performance, across different object and texture categories. None of the evaluated methods performed consistently across different object and texture classes. Object categories were best classified using autoencoders [25], with L2 loss. Similarly, this method performed the best on the segmentation task, but with the structured similarity (SSIM) [25] as the reconstruction loss. This benchmark also nicely represents the level of the generalization performance, especially with the AnoGAN method [16] and its non-competitive results in comparison with the state-of-the-art results in medical domain.

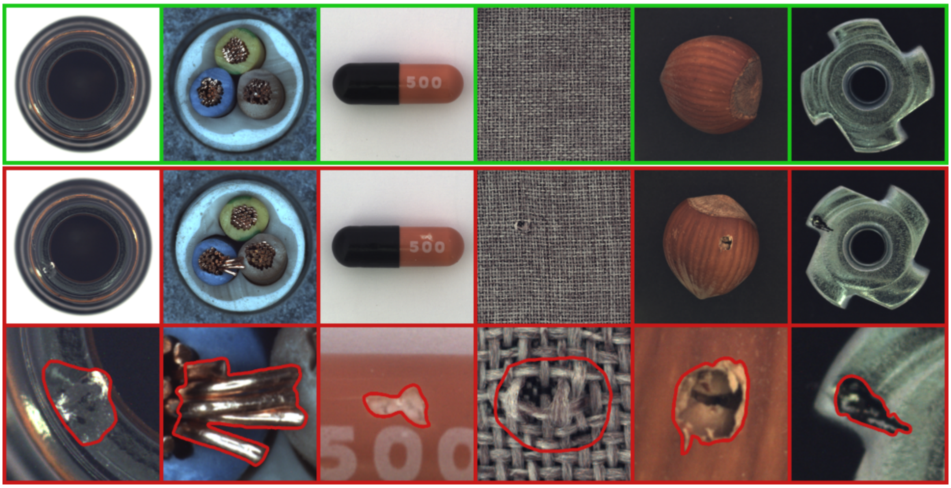
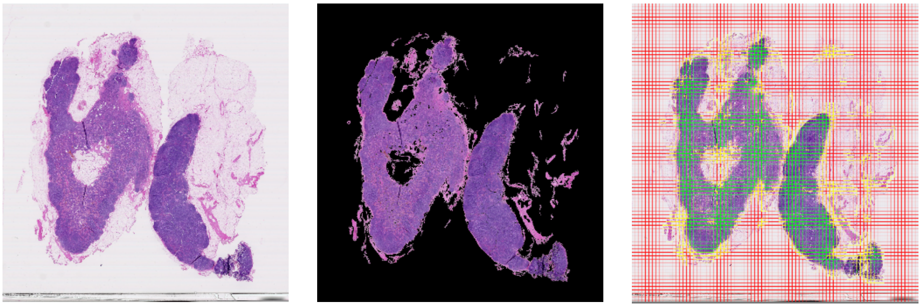


Figure 1: Samples from MVTec AD dataset [21]. First row represents normal samples from the training set (bottle, cable, capsule, carpet, hazelnut, metal nut), second row the same objects with various defects and the third row presents pixel-wise annotations for defective samples. Image adapted from [21].

2) Detection of Lymph Node Metastases in Women with Breast Cancer: Advances in tissue digitalization and in slide scanning technology have opened the possibilities for computer-aided diagnostics to detect cancerous metastases in stained tissue sections. Digital pathology is a new emerging field, utilizing computerized analysis of histopathological images. Breast cancer is just one of the many cancers, where the extent of it is measured by histopathological analysis. Detecting metastases in such gigapixel imagery is prone to error and a time consuming process, where pathologists would benefit greatly by recent advances in computer vision domain.

A competition was organized in 2015 [3] in order to evaluate the machine learning based methods against pathologists. In the challenge setting, some deep-learning based methods achieved better results than a panel of 11 pathologists. 399 whole-slide images were collected from 399 patients at 2 hospitals in the Netherlands. All metastases in the slides where annotated by trained pathologists on the slide level (contains metastases or not - for the slide level classification), as well as separate metastases (for segmentation purposes). The set of images was randomly divided into train (n = 270) and test set (n = 129). All the data is publicly available on the competition websites[1][2]. The first task was designed to evaluate the detection of separate metastases and evaluate the performance against the reference annotations, provided by the pathologists. Free-response Receiver Operator Characteristics curve (FROC) was used to evaluate the performance, at 6 predefined false positive rates. FROC curve shows the true-positive fraction vs. the mean number of false-positive detections in metastasis-free slides only. The goal of the second task was to evaluate the discrimination performance on the whole-slide level. Area Under Curve (AUC) was used for evaluation against pathologists, with and without any time-constraint.

Out of 32 submitted methods, 25 used deep convolutional neural networks (CNNs) and overall performed significantly better compared to traditional approaches. However, preprocessing (e.g. standardizing stain variations, different sampling strategies - class imbalance problem) and augmentation procedures proved to play an important role, compared to the selection of the CNN architecture. After the competition, another approach was presented [29], which improves the competition results significantly and is presented in detail in the next section. All the solutions approached the problem in a supervised fashion, as a patch classification problem. We will approach this problem as an unsupervised visual anomaly detection problem instead. Same data was already utilized in a weakly-supervised fashion using Multiple Instance Learning (MIL) approach [6], utilizing only whole-slide level annotations, presented in section 4.2.



(a) Original WSI (b) Filtered WSI (c) Patches from WSI

Figure 2: Preprocessing of original WSI presented in a) consists out of filtering tissue sections b) and extracting patches c), based on tissue percentage (green ≥ 90%, red ≤ 10% and yellow in-between). Best viewed in digital version with zoom.

[1] https://camelyon16.grand-challenge.org/

[2] https://camelyon17.grand-challenge.org/

4 Methods

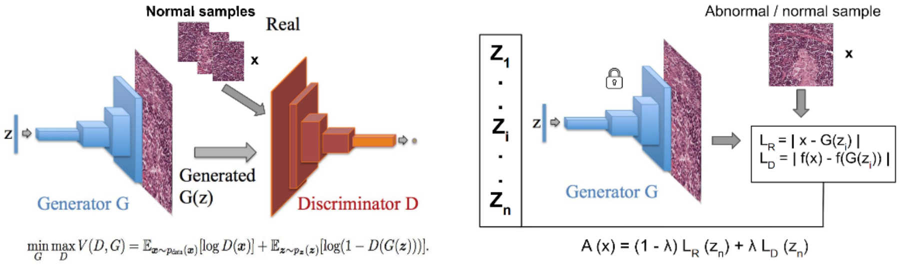
In this section we review representative methods for anomaly detection, based on availability of the data. We focus on the medical domain, specifically on metastases detection from histopathological images. This particular domain represents a challenging task, that has not been considered directly as an anomaly detection problem. This particular problem has been considered in a supervised setting, as well as recently in a weakly-supervised fashion. Unsupervised approaches have not been considered yet. We present these existing approaches, as well as describe current state-of-the-art UAD approaches and present initial results, that demonstrate feasibility to apply them to the domain of metastases detection. The same methods are also applicable to other domains, especially industrial domain, which was presented in this paper; and representative methods for UAD described in this section have also been applied to that domain.

4.1 Supervised Anomaly Detection Methods

Winners of the Camelyon Grand Challenge 2016 [3] on detection of lymph node metastases presented their winning supervised based approach in a technical report [30]. Majority of digitized Whole Slide Image (WSI) consists of background white space, which needs to be segmented, to reduce computational time. Winners first utilized Otsu's algorithm [31] in HSV color space in order to generate segmentation masks. A simple filtering based on the green channel value can also be used, due to the purple and pink tones, resulting from H&E staining. Morphological operators are also applied to remove small objects and artifacts. Results of tissue filtering and patch extraction are presented in Figure 2. We color-coded extracted patches based on the tissue percentage, in order to extract only the patches with sufficient amount of tissue. Metastasis detection framework was then proposed, consisting of patch-based classification part, which produces heatmaps, that are later on processed to obtain WSI-level and lesion-level labels.

Authors utilized GoogLeNet [32] as their best performing CNN architecture. Positive and negative 256 x 256 pixel patches were extracted, according to the provided lesion level labels and used to train the binary classification model. An additional model was learned on hard-negative examples, based on the initial model. The best results were obtained with the highest 40x WSI magnification. Learned models were applied in a sliding window fashion (overlapping patches), to obtain tumour probability maps. For lesion based detection, connected components were identified using the first model, which results were later averaged with the model learned on hard negative examples. For slide-level classification, 28 different geometrical and morphological features were extracted from heatmaps (e.g. percentage of tumour region over whole tissue region). Random Forest classifier was used to discriminate the WSIs with metastases from negative examples. Authors obtained an AUC score of 0.925 for WSI classification and an average FROC score of 0.705. These results showed that close to pathologist-level performance (AUC of 0.966 and FROC of 0.733) can be achieved with supervised deep-learning based models.

Above presented winning solution of the Camelyon 2016 challenge was later further improved by Google [29], by utilizing newer Inception architecture [33], careful image patch sampling and extensive image augmentations. They improved FROC sensitivity score for lesion based detection to 0.885 and AUC score for slide level classification to 0.986, though the evaluation protocol seems not to be exactly the same. They also show that statistically the same slide level classification performance can be achieved solely by using maximum value from the heatmap, instead of handcrafted features and Random Forest classifier.


Normal samples
Real
x
z
Generated
G(z)
Generator G
Discriminator D
$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$.
Abnormal / normal sample
x
Z_1
Z_i
Z_n
z
Generator G
$L_R = | x - G(z_i) |$
$L_D = | f(x) - f(G(z_i)) |$
$A(x) = (1 - \lambda) L_R(z_n) + \lambda L_D(z_n)$

(a) AnoGAN GAN training (b) AnoGAN anomaly detection

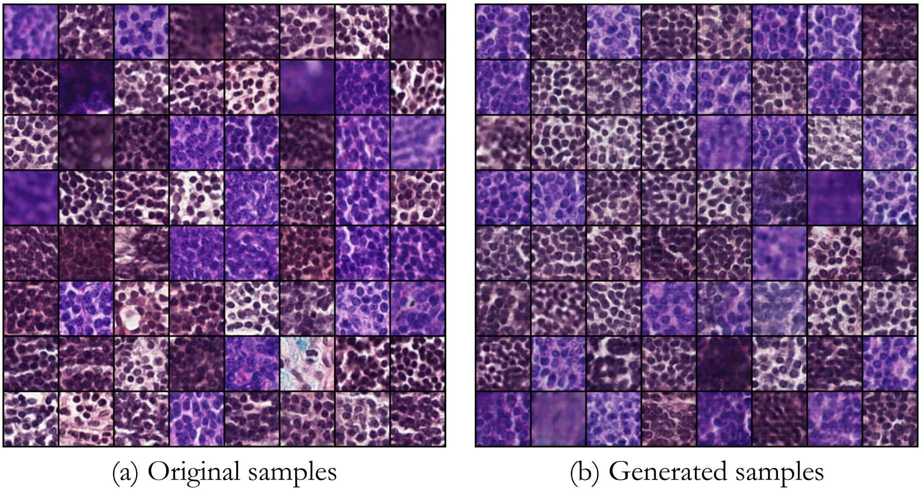
Figure 3: AnoGAN method [16] consisting of DCGAN training a) and iterative optimization procedure b) to find an optimal latent vector for anomaly detection. Image adapted from [16] for digital pathology.

4.2 Weakly-supervised Anomaly Detection Methods

Supervised approaches require abundance of labeled data, which is particularly severe in digital pathology, where digitization of glass slides is expensive, and pixel-level manual labels are time-consuming to obtain, due to gigapixel large pathology imagery. In [6] the authors present a weakly supervised approach, that only utilizes image level reported diagnosis as labels for training, omitting the need for expert pixel-wise annotations. Such a procedure can capture a much wider variance of clinical samples that is not captured in small supervised datasets. They collect large-scale pathology imagery (WSIs) from 1) prostate cancer (prostatic carcinoma), 2) skin cancer (basal cell carcinoma) and 3) breast cancer (axillary lymph nodes), together with slide-level diagnosis, obtained from electronic health records.

With negative slide-level diagnosis, one can be sure, that all the tiles within a negative WSI are negative, not containing the metastases or tumor. On the other hand, with a positive slide-level diagnosis, we know, that at least one tile is positive. This kind of classification problem is a classical formulation of Multiple Instance Learning (MIL), where training instances are arranged in sets, called bags, and a label is provided for the entire bag [34]. Solving MIL task induces the learning of a tile-level representation that can linearly separate the discriminative tiles in positive slides from all other tiles [6]. This is implemented on a tile-level using standard CNN based architectures (e.g. Resnet34) and probability is obtained for each of the tiles of being positive. The top ranked tile (or *K* top ranked) are selected and compared with slide-level ground truth labels, used in cross-entropy loss. In this way, weakly supervised tile-level classifier is learned, that is applied in a similar fashion as in [3]. They used handcrafted features from the obtained heatmaps and learned a Random Forest classifier for slide-level classification, similarly as in [13]. Additionally, they noticed the drawback of such handcrafted aggregation methods for slide-level classification and proposed a new Recurrent Neural Network (RNN) based model that uses features, learned during tile-level classification training.

The performance of the proposed weakly supervised method was evaluated on in-house data, that is not publicly available. They also compared the method with fully supervised approach on Camelyon 2016 challenge data [3]. They implemented a modified supervised winning approach from [30], trained on Camelyon data and evaluated the approach on their in-house data, to evaluate the generalization performance. They noticed a 20% drop in AUC score (from test results on Camelyon data). In comparison, they evaluated their proposed weakly supervised MIL-RCNN method, trained on large-scale in-house data, on Camelyon test set and noticed only 7% drop in AUC score (from test results on in-house data). Unfortunately, they do not report the results of their proposed method, when trained only on Camelyon data.



(a) Original samples

(b) Generated samples

Figure 4: a) Original patches, extracted from histology image and b) generated artificial patches from DCGAN [35] based GAN network, as used in AnoGAN method [16].

4.3 Unsupervised Anomaly Detection Methods

In comparison with supervised and weakly-supervised approaches, unsupervised approaches omit the need for expertly labeled data. UAD is a relatively new domain and has seen particular improvements and rise of interest with the introduction of deep generative methods. In this section we introduce two main approaches, one based on GANs [36] and the other one based on autoencoders [37]. None of the approaches has been applied to challenging digital pathology imagery. Besides introduction to the methods, we also present preliminary results, that demonstrate feasibility to apply presented methods for detection of cancerous regions in histology imagery.

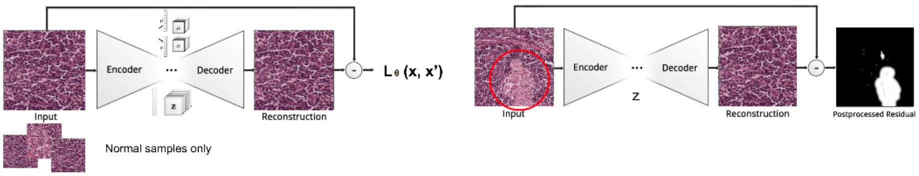
1) GAN based UAD methods: AnoGAN method [16], presented in Figure 3, represents the first work, where GANs are used for anomaly detection in medical domain. A rich generative model is constructed on healthy examples of optical coherence tomography images of the retina and a methodology is presented for image mapping into the latent space, to generate the closest example to the presented query image, to be able to detect and segment the anomalies in an unsupervised fashion. Given a set of healthy images, smaller patches were extracted and used to train a generative model, based on the DCGAN [35] architecture, in order to learn the manifold of healthy examples. In this way, the model captures the variability of the training examples in an unsupervised fashion. Labels are only given during the testing, to evaluate the detection performance.

GANs consists of generator (*G*) and discriminator part (*D*). The generator *G* learns a mapping *G(z)*, where *z* represents a sampled 1D vector from the uniformly distributed input noise, sampled from the latent space - consisting of healthy examples. Discriminator on the other hand, maps an input 2D image to a scalar value, representing the probability of the input being a real image, sampled from the training data, or a generated one - produced by *G(z)*. *G* and *D* are trained in an alternating fashion, using a two-player minimax game. The discriminator *D* is trained to maximize the probability to discriminate the real image, from the generated one. Generator (*G*) is on the other hand trained to fool the discriminator. After the adversarial training is completed, the generator learns how to generate realistically looking healthy examples, captured in the training set. When query image *x* is presented, to detect the anomaly, the goal is to find the closest point *z* in the latent manifold of healthy examples. This is done in an iterative fashion from a randomly sampled initial latent vector z_1, which is updated back using backpropagation in $i = 1,2,..,n$ steps, via residual (L_R) and discrimination loss (L_D), to obtain the optimal latent vector z_n (only the coefficients of *z* are modified, *G* and *D* parameters are kept fixed). Residual loss captures similarity of the query image to the generated one $G(z_i)$, while discrimination loss ensures that the generated image $G(z_i)$ lies on the learned manifold of healthy training examples. The mapping and corresponding losses are inspired by the work of semantic image inpainting using GANs [38], which poses a similar problem setup. The combined residual and discrimination loss for z_n can be directly used as an anomaly score $A(x)$ and the resultant residual image between $G(z_n)$ and query image, for pixel-wise anomalous region segmentation. The whole process of training AnoGAN method [16] is visually presented in Figure 3.

Iterative optimization approach to find the optimal latent vector is time-consuming and not applicable for real-time anomaly detection. Recently presented f-AnoGAN method [8] greatly improves inference times, at a similar performance rate, by replacing iterative optimization approach with a trained encoder mapping from images to corresponding location in the learned latent space. Besides, training GANs can be a very unstable process and mostly smaller resolution images are used. AnoGAN and f-AnoGAN methods utilize baseline DCGAN [35] and Wasserstein GAN (WGAN) [39] architectures and do not consider recent works, that are able to generate higher resolution images in a more stable way [40], [41]. Capability to generate realistically looking histology imagery is crucial, in order to generate accurate cellular structure. We present baseline results of the DCGAN [35] architecture in Figure 4. These initial results with a baseline method, that was also used in AnoGAN method, demonstrate the applicability of such methods to digital pathology domain.

2) AE based UAD methods: Above presented UAD methods are modelling normal samples with GANs. Autoencoder (AE) based methods are one of the simplest and first approaches, that are also used for visual anomaly detection, by learning how to reconstruct the input image through a bottleneck, via encoder (*E*) and decoder (*D*) networks. Generative AEs (i.e. variational autoencoders [37]) were also introduced and used in a recent UAD work for lesion detection in brain MR images [9]. AE based method are trained in a self-supervised way, such that they learn how to reconstruct input training images. This is achieved by mapping an input to a bottleneck, which can in fact be a distribution or a direct mapping. When introduced with normal samples only, they learn how to reconstruct such normal samples and in the case of VAE, they are also able to generate them, similarly to GANs. When we introduce anomalous sample, the method is able to reconstruct it, the way that the normal sample should look like. We are then able to threshold the reconstruction error, in order to detect the anomaly, as well to segment them, by computing a residual image. This process is visually presented in Figure 5, the way, that the method would be used in digital pathology setting.

GANs are known to produce very sharp images, due to adversarial training, but are having issues with stable training and mode collapses, which results in learning to generate just a few examples [35]. GAN and VAE concepts have been recently combined into VAEGAN framework [42], combining the best of the two approaches. Adding an adversarial loss and discriminator to the AE/VAE framework forces the decoder to generate better reconstructions, that will fool a discriminator. In [9] they also used spatial VAEs, replacing the mapping to dense 1D bottleneck z with a fully convolutional encoder-decoder network, resulting in a higher dimensional spatial bottleneck z, omitting the loss of the spatial information in the bottleneck encoder function. The presented AnoVAEGAN UAD method was compared against AnoGAN [16] method and variations of AE/VAE architectures with different types of bottlenecks (i.e. dense vs. spatial) and their dimensions. Similar to GAN approaches, no AE based approach has been utilized for anomaly detection in gigapixel histology imagery.



(a) AE based UAD method training (b) AE based anomaly detection

Figure 5: Basic architecture of AE based method for anomaly detection, consisting of AE training a) and AE inference b), resulting to residual image, used for anomaly detection and segmentation. Image adapted from [9] for digital pathology.

5 Conclusion

Visual anomaly detection is an important process in many domains and recent advancements in deep generative based methods have shown promising results towards applying them in an unsupervised fashion. This has sparked research in many domains, that did not benefit much from traditional supervised deep-learning based approaches.

Most of the existing methods are applied to medical domain, where vast amount of imagery data is available, but without any detailed labels to learn state-of-the-art supervised models. All the appearances of anomalies in real-world applications are usually also not known in advance and are as such impossible to label. Benefits of such methods have recently also been recognized in industrial inspection domain, where the need for rapid product development is making the existing supervised approaches inappropriate to use, due to time constraints to collect anomalous samples, as well as wide-range of potential anomalies that can occur and are unknown in advance. The presented UAD methods have been developed and evaluated on particular limited real-world domains or even on existing classification datasets and significant performance drops are visible when applied to other domains. This has been seen through several presented works, that evaluated the existing UAD methods, along with the newly presented ones, on new application domains. Another important issue is the robustness of existing UAD methods to contaminated training data. Existing UAD methods are not really unsupervised due to the requirement that completely anomaly-free data is available for training the methods, therefore implicitly implying the need for weak labelling.

UAD in visual data is a relatively new domain, that has seen particular improvements with the introduction of generative based methods. Unprecedented amount of visual data that is captured every day in different domains represents an untapped potential for unsupervised based methods, that will be able to leverage this data as it is. Addressing the issues of current UAD approaches will enable their wider usage, especially in data-heavy domains. Dual-use of UAD approaches that enables novelty detection can also represent a major diagnostic tool for early cancer detection and rare disease detection, thereby support the development and evaluation of personalized medicine, and thus address a much wider societal challenge.

Acknowledgment

This work was partially supported by the European Commission through the Horizon 2020 research and innovation program under grant 826121 (iPC).

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online].
Available: http://doi.acm.org/10.1145/1541880.1541882

[2] R. Chalapathy and S. Chawla, “Deep Learning for Anomaly Detection: A Survey,” ArXiv, vol. abs/1901.03407, 2019.

[3] B. Ehteshami Bejnordi, M. Veta, P. Johannes van Diest, B. van Ginneken, N. Karssemeijer, G. Litjens, J. A. W. M. van der Laak, and the CAMELYON16 Consortium, “Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer,” JAMA, vol. 318, no. 22, pp. 2199–2210, 12 2017. [Online]. Available: https://doi.org/10.1001/jama.2017.14585

[4] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-Based Deep-Learning Approach for Surface-Defect Detection," Journal of Intelligent Manufacturing, May 2019. [Online]. Available: https://doi.org/10.1007/s10845-019-01476-x

[5] M. M. R. Siddiquee, Z. Zhou, N. Tajbakhsh, R. Feng, M. B. Gotway,Y. Bengio, and J. Liang, "Learning Fixed Points in Generative Adversarial Networks: From Image-to-Image Translation to Disease Detection and Localization," in The IEEE International Conference on Computer Vision (ICCV), October 2019.

[6] G. Campanella, M. G. Hanna, L. Geneslaw, A. Miraflor, V. W. K. Silva, K. J. Busam, E. Brogi, V. E. Reuter, D. S. Klimstra, and T. J. Fuchs, "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images," Nature medicine, vol. 25, no. 8, pp. 1301–1309, 2019.

[7] P. Courtiol, E. W. Tramel, M. Sanselme, and G. Wainrib, "Classification and disease localization in histopathology using only global labels: A weakly-supervised approach," arXiv preprint arXiv:1802.02212, 2018.

[8] T. Schlegl, P. Seebock, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks," Medical Image Analysis, vol. 54, pp. 30 – 44, 2019. [Online].
Available: http://www.sciencedirect.com/science/article/pii/S1361841518302640

[9] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images," in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, A. Crimi, S. Bakas, H. Kuijf, F. Keyvan, M. Reyes, and T. van Walsum, Eds. Cham: Springer International Publishing, 2019, pp. 161– 169.

[10] L. Beggel, M. Pfeiffer, and B. Bischl, "Robust Anomaly Detection in Images using Adversarial Autoencoders," ArXiv, vol. abs/1901.06355, 2019.

[11] A. Berg, J. Ahlberg, and M. Felsberg, "Unsupervised Learning of Anomaly Detection from Contaminated Image Data using Simultaneous Encoder Training," ArXiv, vol. abs/1905.11034, 2019.

[12] F. D. Mattia, P. Galeone, M. D. Simoni, and E. Ghelfi, "A Survey on GANs for Anomaly Detection," 2019.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[15] L. Ruff, R. A. Vandermeulen, N. Gornitz, A. Binder, E. M¨uller, K.- R. Muller, and M. Kloft, "Deep Semi-Supervised Anomaly Detection," 2019.

[16] T. Schlegl, P. Seeb¨ock, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," in Information Processing in Medical Imaging, M. Niethammer, M. Styner, S. Aylward, H. Zhu, I. Oguz, P.-T. Yap, and D. Shen, Eds. Cham: Springer International Publishing, 2017, pp. 146–157.

[17] Emeršič, D. Štepec, V. Štruc, and P. Peer, "Training Convolutional Neural Networks with Limited Training Data for Ear Recognition in the Wild," in 2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017), May 2017, pp. 987–994.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, 05 2012.

[20] E. A. Donahue, T.-T. Quach, K. Potter, C. Martinez, M. Smith, and C. D. Turner, "Deep learning for automated defect detection in high-reliability electronic parts," in Applications of Machine Learning, M. E. Zelinski, T. M. Taha, J. Howe, A. A. S. Awwal, and K. M. Iftekharuddin, Eds., vol. 11139, International Society for Optics and Photonics. SPIE, 2019, pp. 30 – 40. [Online]. Available: https://doi.org/10.1117/12.2529584

[21] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD - A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in CVPR, 2019.

[22] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," Special Lecture on IE, vol. 2, no. 1, 2015.

[23] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," arXiv preprint arXiv:1802.06360, 2018.

[24] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Muller, and M. Kloft, "Deep one-class classification," in International Conference on Machine Learning, 2018, pp. 4393–4402.

[25] P. Bergmann, S. L¨owe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," arXiv preprint arXiv:1807.02011, 2018.

[26] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly detection in nanofibrous materials by cnn-based self-similarity," Sensors, vol. 18, no. 1, p. 209, 2018.

[27] T. Bottger and M. Ulrich, "Real-time texture error detection on textured surfaces with compressed sensing," Pattern Recognition and Image Analysis, vol. 26, no. 1, pp. 88–94, 2016.

[28] C. Steger, M. Ulrich, and C. Wiedemann, Machine vision algorithms and applications. John Wiley & Sons, 2018.

[29] Y. Liu, K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado et al., "Detecting cancer metastases on gigapixel pathology images," arXiv preprint arXiv:1703.02442, 2017.

[30] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," arXiv preprint arXiv:1606.05718, 2016.

[31] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, Jan 1979.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Computer Vision and Pattern Recognition (CVPR), 2015.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[34] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," Pattern Recognition, vol. 77, pp. 329–353, 2018.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2014.

[38] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5485–5493.

[39] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in Proceedings of the 34th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 214–223.

[40] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.

[41] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.

[42] J. Donahue, P. Krahenbuhl, and T. Darrell, "Adversarial feature learning," arXiv preprint arXiv:1605.09782, 2016.

DOPOLDANSKA SEKCIJA

Industrijske aplikacije
Medicinske in biomedicinske aplikacije
Drugo
Študentske aplikacije

KORAKI PRIPRAVE 3D NATISNJENEGA ŽILNEGA MODELA ZA NAMEN NAČRTOVANJA OPERACIJE, Z UPORABO MR IN CT SLIK PACIENTA Z MOŽGANSKO ANEVRIZMO

ANDREJ VOVK
Univerza v Ljubljani, Medicinska fakulteta, Center za klinično fiziologijo, Ljubljana, Slovenija, e-pošta: andrej.vovk@mf.uni-lj.si

Povzetek Za mikrokirurški način zdravljenja anevrizme je 3D predstavitev patologije in njene okolice lahko v veliko pomoč pri načrtovanju operacije. Slike možganov in ožilja lahko pridobimo z neinvazivnim MR slikanjem. Običajno se v kliniki uporabi tudi CT slikanje, ki pri pripravi 3D modela omogoča še prikaz lobanje. Obdelava slik se začne s poravnavo posameznih 3D slik na T1 strukturno sliko. Nato vsako posamezno 3D sliko segmentiramo na način, da iz nje pridobimo želeno strukturo. Naslednji korak je pretvorba posameznih segmentiranih slik oz. želenih 3D volumenskih struktur v ploskovne modele. Pripravljene ploskovne modele vseh izbranih struktur nato združimo in jih dodatno obdelamo, ter po želji obarvamo, da čim bolje predstavimo patologijo oz. področje zanimanja. Čeprav je že 3D predstavitev z možnostjo rotacije na računalniku v veliko pomoč pri predstavi struktur, je v primerih načrtovanja operacij zelo uporabna tudi fizična predstavitev modela, ki jo dobimo s 3D tiskanjem.

Ključne besede:
možganske anevrizme,
3D prikaz,
3D tiskanje,
načrtovanje operacije,
mikrokirurško zdravljenje.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1.2
ISBN 978-961-286-337-1

1 Uvod

Anevrizma je izraz za izbočenje žilne stene na razcepiščih arterij, ki nastane zaradi njene oslabelosti. Žilna stena ob prevelikem raztezanju poči, kar povroči možgansko krvavitev. Ob ugotovitvi take patologije je zato nujno zdravljenje oz. operacija. Seveda je nujnost odvisna od velikosti in lokacije anevrizme. Prav tako je pristop zdravljenja odvisen od velikosti anevrizme. Pri večjih anevrizmah se uporabi mikrokirurško izključevanje anevrizme iz krvnega obtoka s kirurško sponko. Manjše anevrizme lahko zdravimo z endovaskularnim načinom, ki je manj invaziven. Mikrokirurški pristop pomeni, da nevrokirurg odpre lobanjo ter možgansko ovojnico ter previdno pristopa do anevrizme, očisti okolico anevrizme in namesti kirurško sponko. Glede na lokacijo anevrizme in preplet žil v okolici, je lahko operacija zelo zahtevna, že za izkušene kaj šele za neizkušene nevrokirurge. Z nevrokirurškim oddelkom na UKC v Ljubljani smo ugotovili, da v zahtevnejših primerih možganskih patologij kot je anevrizma, predstavlja 3D vizualizacija in fizična predstavitev dejanskega stanja veliko dodano vrednost pri načrtovanju operacije. Na 3D modelu se lahko bolje predvidi pristop do mesta patologije kot tudi uporaba primernega kirurškega orodja in sponk.

2 PREDSTAVITEV PROBLEMA IN PRIČAKOVANE REŠITVE

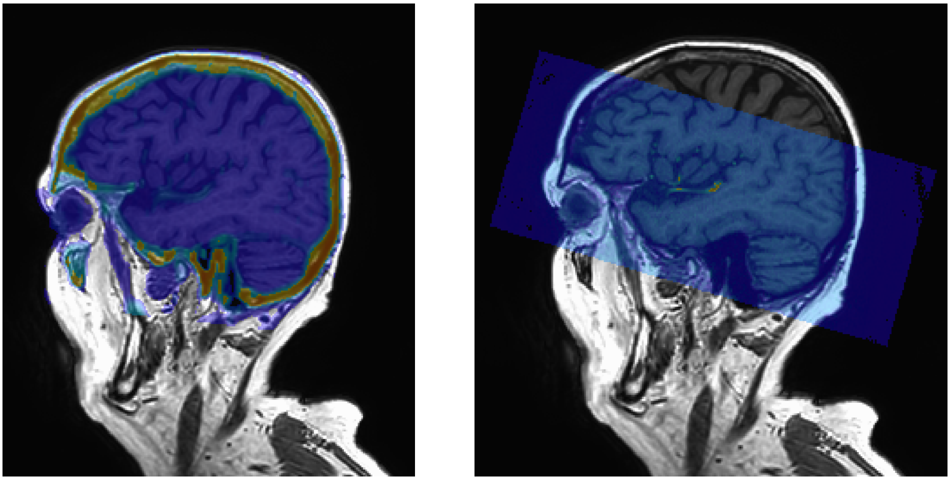
2.1 Zajem slik

Za slikanje žil uporabimo neinvazivno brezkontrastno MR TOF sekvenco (*three-dimensional high-resolution time-of-flight MR angiography acquisition)*; na Philips MR tomografu z imenom s3DI_MC_HR. Ta sekvenca omogoča vizualizacijo pretoka krvi po arterijskih žilah in posredno s tem dobimo potek arterijskih žil. Parametri TOF sekvence so: velikost rekonstruiranih vokslov 0,34mm x 0,34mm x 0,5mm pri velikosti matrike 560 x 560 x 210 (zajemanje je bilo s pol manjšo resolucijo in s presledkom -0,5mm). Čas trajanja zajema sekvence je 4,5min. Za bolj natančen prikaz možganskih struktur moramo posneti tudi T1 in T2 obteženo strukturno sliko možganov. (tipični raziskovalni sekvenci: 236 sagitalnih rezin, matrika = 336×336, dimenzija vokslov = 0,7mm × 0,7mm × 0,7mm; **T1**: TE = 5,7 ms; TR = 12 ms; flip angle = 8°; čas snemanja ~6min ; **T2**: TE = 414 ms; TR = 2500 ms; flip angle = 90°; čas snemanja ~3min). Če želimo ob možganih in žilah prikazati tudi lobanjo, moramo uporabiti CT slikanje. CT slikanje lahko zaradi invazivnosti omejimo le na rezine okoli možnih pristopov za operacijo (čas snemanja ~1min).

2.2 Obdelava slik

Med uporabniki - raziskovalci na področju nevro znanosti se za obdelavo slik največkrat uporabljajo odprtokodni paketi kot so *FSL [1]*, *AFNI [2]*, *Freesurfer [3]*, ter *3DSlicer [4]*. V nadaljevanju bodo iz omenjenih paketov predstavljeni programi za posamezne korake.

Za poravnavo 3D slik obstaja veliko orodij, ki so običajno del večjih programskih paketov. Ker med sabo poravnavamo slike istega preiskovanca, zadostuje uporaba linearne poravnave, imenovane tudi rigidna (6 DOF) transformacija, običajno z uporabo MMI matrike (Mattes Mutual Information cost matrix) ali lokalne korelacijske funkcije; prikazano na sliki 1. V programu *3DSlicer* lahko za tovrstno poravnavo uporabimo modul *General Registration* (BRAINS), v *FSL* paketu to omogoča funkcija *flirt*, v programskem paketu *AFNI* za to uporabimo *align_epi_anat.py* program. Seveda tudi v *Matlab*u in v *Python*u, ki sta bolj splošni orodji za najrazličnejše obdelave podatkov, najdemo knjižnice za poravnave oz. registracije slik.

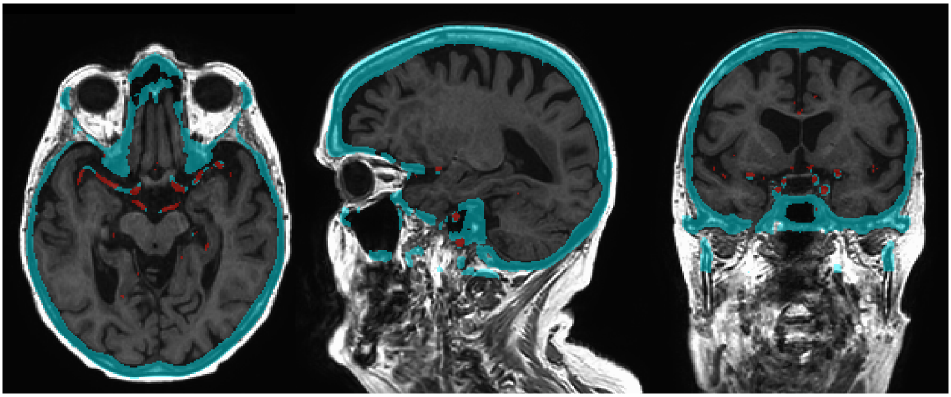


Slika 1: Levo poravnana TOF slika na T1 obteženo strukturno sliko, na desni poravnana CT slika na T1 sliko.

Glede na preferenco uporabe programskih orodij za slikovno obdelavo, je segmentacija možganov že med paketi razdeljena na metode z uporabo standardnih možganov ali brez teh predlog. Če obdelujemo možgane brez prevelikih patologij, daje metoda z uporabo standardnih možganov kot jo uporablja paket Freesurfer najboljše rezultate. Možgane na ta način lahko segmentiramo ne samo na belino in sivino, ampak na še podrobnejše strukture. Segmentacija s Freesurferjem sicer traja več kot 10ur. Procesorsko manj zahtevne variante segmentacije vsebujeta paketa AFNI (3dSeg) in FSL (FAST).

TOF in CT slike lahko, zaradi dovolj poudarjenih specifičnih struktur že z metodo zajemanja slike, segmentiramo z enostavno "threshold metodo" - določanje praga sivin. Pred tem je smiselno določiti območje opazovanja (ROI – region of interest), da se izognemo kasnejšemu filtriranju nekoristnih informacij na prevelikem področju. Ena od možnosti filtriranja nekoristnih, kar največkrat pomeni premajhnih regij, po določanju praga sivin, je uporaba gručenja (clustering). Na ta način lahko izločimo premajhna področja, ki so v primeru segmentacije lobanje ali ožilja, kot v našem primeru, neka nepovezana in najverjetneje nekoristna področja oz. strukture.

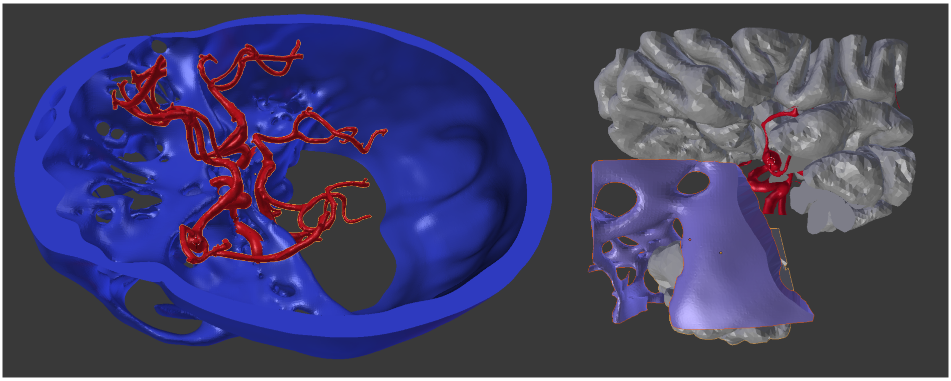
Pred pretvorbo segmentiranih področij v ploskovni model, moramo preveriti in odpraviti morebitna prekrivanja segmentiranih struktur, kar sicer lahko kasneje storimo tudi v ploskovnem modelu, vendar z uporabo surovih anatomskih slik, ki jih lahko naložimo nad segmentirane, bolj točno določimo katera struktura je v spornem področju pravilna, kot je prikazano na sliki 2.



Slika 2: Prikaz prekrivanja žil (z rdečo) ter kosti (turkizno). To se zgodi zaradi občutljivosti CT slikanja tudi na žilne strukture.

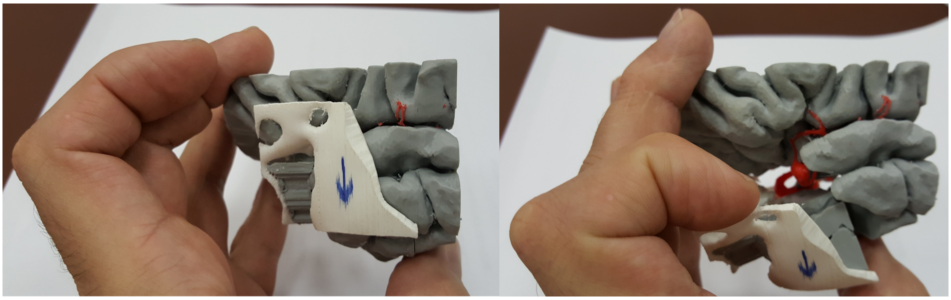
Nadaljni korak pred pretvorbo volumskih struktur v ploskve je glajenje zunanjih vokslov (najmanjši prostorski element). Zaradi lepše pretvorbe je potrebno preveriti, da se zunanji/robni voksli stikajo s ploskvami in ne samo z robovi ali vogali (v Freesurferju to storimo z ukazom mri_pretess). Samo pretvorbo izvedemo z ukazom mri_tessellate (v Freesurferju). V AFNI paketu izvedemo pretvorbo v ploskovni model z ukazom 3dVol2Surf. Algoritmi za pretvorbo iz 3D volumetrične slike v 3D ploskovni model izhajajo iz matematičnega algoritma Marching cubes [5] in se seveda najdejo tudi med knjižnicami Pythona in Matlaba.

V program Blender [6] lahko nato uvozimo vse modele in jih popravimo, če je to potrebno. Tu lahko posamezne dele tudi različno obarvamo in razrežemo, da dobimo čim boljšo predstavo o kirurškem pristopu do patologije. Čeprav so v Blenderju prikazane tudi žile s premerom manj kot 0,5mm, moramo razmisliti s kakšno metodo bomo izvedli 3D tiskanje. V primeru tiskanja z nalagalno/ekstruzijsko metodo je bolje omejiti prikaz žil, ki so manjše od 0.7mm ali celo 1mm, oz. glede na podrobnosti, ki nas zanimajo; drugače imamo preveč dela s čiščenjem in ločevanjem podpornega materiala od tankih struktur.



Slika 3: Na levi je v Blenderju predstavljeno arterijsko ožilje z anevrizmo spodaj-levo, omejeno na žile zadostne debeline za 3D tiskanje. Na desni strani je prikazan izsek področja zanimanja z anevrizmo v sredini, ki se nahaja skrita za temporalnim režnjem.

Za 3D tiskanje modelov, prikazanih na sliki 3, smo uporabili tiskalnik sigma BCN3D z natančnostjo printanja 0.08mm ter z dvema brizgalnima šobama, kar omogoča dvobarvno tiskanje oz. ob primernem razrezu lahko prikažemo v končnem sestavljenem modelu tudi več barv. V prikazanem primeru, smo razdelili model na dva kosa, tako da je v enem kosu lobanja natisnjena z belo in del možganov natisnjen s sivo, v drugem kosu so natisnjeni možgani s sivo, ter žile z rdečo; prikazano na sliki 4. Tiskanje obeh kosov je trajalo skoraj 40 ur in pri tem je bilo porabljenega skupaj 94g PLA filamenta (20€/kg).



Slika 4: Slika natisnjenega 3D modela območja anevrizme z možgani. Model je sestavljen iz dveh delov, tako da z odstranitvijo dela temporalnega režnja, vidimo bolj natančno tudi anevrizmo.

Ker je pri 3D tiskanju previsnih predelov potrebno uporabiti podporni material, je zaradi kompleksnih struktur čiščenje dolgotrajno in natančno opravilo (seveda odvisno od zahtevane estetike končnega modela) min 2 uri. Pri tem je glavno orodje skalpel.

3 Zaključek

Naj za zaključek povzamem še vse stroške izdelave 3D modela anevrizme za načrtovanje kirurškega posega. Glede na to, da so slikanja s CT in MR tomografom že del kliničnih preiskav, lahko rečemo, da sama slikanja ne predstavljajo dodatnih stroškov. Po trenutnih cenah na samoplačniškem trgu, moramo za MR slikanje glave odšteti okoli 220€ in za CT slikanje okoli 150€.

Za 3D tisk je cena porabljenega materiala za predstavljeni model okoli 2€. Če pri tem upoštevamo še porabo elektrike (40ur x 300W x 0,17€/kWh cca 2€), skupaj za 3D tiskanje porabimo 4€.

Računalniško delo segmentiranja, ter priprava in razrez modela za 3D printanje lahko časovno ocenimo na približno 10ur (seveda odvisno od izkušenj, ter od zahtevane kvalitete modela).

Z boljšimi metodami slikanja bi lahko zajeli tudi manjše žile z MR slikanjem [7], seveda bi za tehniko tiskanja morali nato izbrati 3D tiskanje z laserskim topljenjem materiala v prahu.

Literatura

[1] S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, and P.M. Matthews (2004), Advances in functional and structural MR image analysis and implementation as FSL, NeuroImage, vol. 23, str. 208-219. Program dostopen na: https://fsl.fmrib.ox.ac.uk

[2] R.W. Cox (1996), AFNI: Software for Analysis and Visualization of Functional Magnetic Resonance Neuroimages, Computers and Biomedical Research, vol. 29, str. 162-173. Program dostopen na: https://afni.nimh.nih.gov

[3] Dale, A.M., Fischl, B., Sereno, M.I. (1999), Cortical surface-based analysis. I. Segmentation and surface reconstruction, Neuroimage, vol. 9, str:179-194, 1999. Program dostopen na: https://surfer.nmr.mgh.harvard.edu

[4] Gering D.T., Nabavi A., Kikinis R., Grimson W.E.L., Hata N., Everett P., Jolesz F.A., Wells III W.M. (1999), An Integrated Visualization System for Surgical Planning and Guidance using Image Fusion and Interventional Imaging, Int Conf Med Image Comput Comput Assist Interv., vol. Sep, str. 809-828.

[5] Lorensen WE, Cline HE (1987), Marching cubes: A high resolution 3D surface construction algorithm, ACM Computer Graphics, vol 21, str, 163–169.

[6] Community, B.O. (2018), Blender - a 3D modelling and rendering package, Stichting Blender Foundation, Amsterdam. Program dostopen na: http://www.blender.org.

[7] Thomas W. Okell, Meritxell Garcia, Michael A. Chappell, James V. Byrne, Peter Jezzard (2019), Visualizing artery-specific blood flow patterns above the circle of Willis with vessel-encoded arterial spin labeling, Magn. Reson. Med., vol. 81, str. 1595–1604.

PREPROCESSING TECHNIQUES FOR HETEROGENEOUS FACE RECOGNITION WITH OFF-THE-SHELF DEEP NETWORKS

NEJC PRESEČNIK, VITOMIR ŠTRUC & KLEMEN GRM

University of Ljubljana, Faculty of Electrical Engineering, Slovenia, e-mail:
vitomir.struc@fe.uni-lj.si, klemen.grm@fe.uni-lj.si

Abstract The Heterogeneous Face Recognition (HFR) consists of matching face images captured in different imaging domains, for example, visible light (VIS) images to near-infrared (NIR) images. In comparison to face recognition with VIS images only, attempting recognition with NIR images using common pretrained deep convolutional neural networks (DCNNs) usually results in suboptimal performance, since most DCNNs are trained solely on VIS image data. In this paper we investigate whether it is possible to improve HFR with existing DCNN models trained explicitly for VIS-image recognition through simple preprocessing techniques. The idea here is to reduce the difference in visual appearance of the two types of images caused by imaging faces in different modalities through preprocessing. To this end, we evaluate nine different preprocessing techniques ranging from simple contrast enhancement methods and photometric normalization approaches to gradient operators and texture descriptors and assess their impact on HFR performance in conjunction with the popular VGGFace DCNN. Preprocessing techniques are evaluated on public datasets CARL and NIVL. Results show that it is possible to achieve performance improvements through preprocessing. Our findings suggest that there is ample room for further research.

Keywords:
face recognition, near-infrared, image processing, deep learning, VGGFace architecture.

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-337-1.4
ISBN 978-961-286-337-1

1 Introduction

Heterogeneous face recognition (HFR) refers to the problem of matching faces across different visual domains [1]. The main challenge involved in HFR is to overcome the semantic gap between face images captured with different imaging devices (e.g., visual light (VIS), near-infrared (NIR) or 3D devices), in different quality settings (e.g., high-resolution, low-resolution images), or between images produced by an artist or generated automatically by an imaging sensor (e.g., forensic sketches *vs.* digital images). HFR has seen increased interest in recent years mainly due to the vast number of potential applications in security and law enforcement. For example, VIS-NIR matching is important in biometric security control, where enrollment images are commonly captured in controlled conditions under visual light, whereas probes are typically taken by security cameras in the infrared spectrum. Sketch-based recognition, on the other hand, is crucial for law-enforcement, where eyewitness sketches need to be matched against mugshot datasets to identify suspects [1].

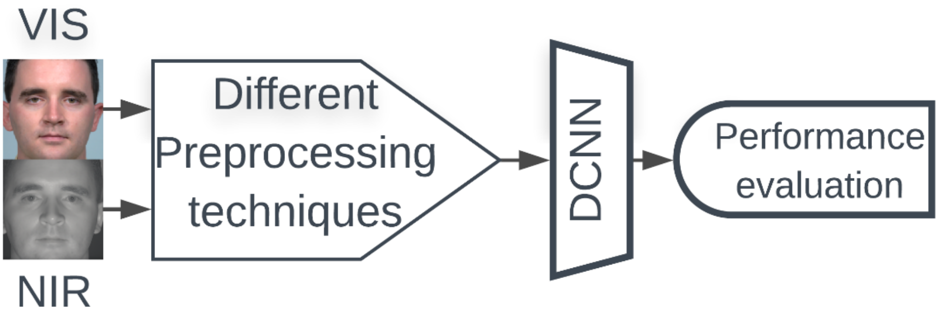


Figure 1: In this paper we investigate whether simple preprocessing techniques can help improve performance for VIS-NIR heterogeneous face recognition (HFR) in an experimental setup, where an existing DCNN pretrained for recognition on VIS images only is used for recognition on VIS images only is used for descriptor computation. To explore this issue, we study the impact of 9 different preprocessing techniques in conjunction with the VGGFace model.

There are three main families of solutions to the HFR problem. The first is the use of features that are invariant to the change in modality [2] [3] [4]. The second is translating images of one modality to the other artificially [5] [6] [7] and the third is projection of both modalities of face images to a common subspace [8] [9]. A large amount of work focuses on the first solution, i.e., the use of invariant features. These features can be either hand-crafted or learned from data, as seen with most recent deep learning methodologies. Despite substantial progress in the field of HFR, the performance is not yet comparable to face recognition performance reported with only VIS images, where DCNNs are currently dominating the field. Since DCNNs require a large amount of data for training, it is challenging to train these on HFR datasets, which typically are of rather modest size in comparison to datasets for VIS-based face recognition. Researchers, thus, have to resort to alternative solutions, including transfer learning, domain adaptation, image synthesis and others.

A straightforward alternative to the outlined methods is to use simple image preprocessing to reduce the modality induced appearance gap and feed the preprocessed images to a powerful pretrained DCNN designed for VIS image face recognition. Such an approach is reminiscent of synthesis (or translation-based) techniques mentioned above but is computationally much simpler as no training is involved. In this paper we are interested whether such preprocessing techniques can in fact be used to improve HFR with an off-the-shelf DCNN trained for face recognition with VIS images. The preprocessing techniques considered mainly follow a typically assumption made in the literature for VIS-NIR matching problems, i.e., that the high-frequency part of the data is similar in both domains. Specifically, nine different techniques are investigated, including contrast enhancement techniques, gradient operators, photometric normalization techniques and texture descriptors, and tested in conjunction with the popular VGGFace model [10]. Verification experiments are performed on original and preprocessed images of two datasets: the CARL and NIVL datasets.

The results of the experiments show that HFR with off-the-shelf DCNN can be improved to some extent, with the use of proper preprocessing techniques, especially contrast enhancement techniques, which seem to be particularly suited for use with pretrained DCNNs.

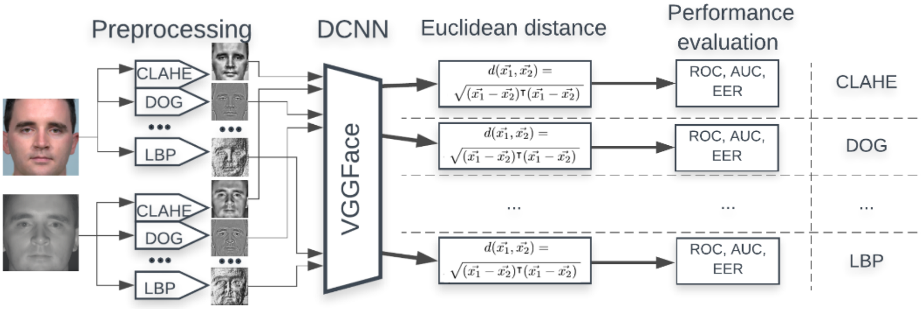


Figure 2: The proposed method in this paper: VIS and NIR images are preprocessed with 9 different preprocessing techniques. For every preprocessing, its output image is inserted in VGGFace DCNN for feature extraction. For verification experiment, Euclidean distance is used between pairs of images. Every pair consists of VIS image and NIR image. The goal is to find out how CLAHE, DOG, LBP, SSR, etc. affect HFR with VGGFace DCNN.

2 Related work

This section discusses prior work relevant to our study. The first part reviews existing models and solutions for heterogeneous face recognition and the second part elaborates on existing DCNN-based approaches to face recognition in the visible spectrum.

2.1 Heterogeneous face recognition

Methods for heterogeneous face recognition (HFR) mainly focus on three strategies to alleviate the modality gap [11].

The first, most common strategy, is designing features that are invariant to the modalities considered, while simultaneously being discriminative for person identity. Gong et al. [12], for example, proposed a new feature descriptor called common encoding model for heterogeneous face recognition, which is able to capture common discriminant information, such that the large modality gap can be reduced at the feature extraction stage. More recently, several feature learning methods were proposed, where features are automatically learned from raw data. These methods different paradigms including coupled subspace learning [13] [14] [3], dictionary learning [4] [15] and deep learning [14] [2] [16].

The second strategy focuses on synthesizing one modality into the other in such way, that they can be compared directly. For instance, Tang and Wang proposed synthesizing face images from pseudo-sketches using an eigen-transformation [6] and multi-scale Markov random fields [1]. Liu et al. [17] proposed to generate sketches from photos using local linear embedding. Gao et al. [7] utilized embedded hidden Markov model to learn the nonlinear relationship between face sketches and photos. The advantage of second strategy is that the synthesized images can be used directly for recognition with VIS images, which by itself gives better performances due to much bigger datasets being available in the VIS domain.

The third strategy aims to project both modalities of face images to a common subspace in which they are more comparable than in the original representation. Lin and Tang [8] propose a new algorithm called Common Discriminant Feature Extraction that relies on eigen-decomposition of a discriminant space common to two domains. Liu et al. [9] propose a new feature set called the Light Source Invariant Features in order to extract invariant representations from near-infrared (NIR) and VIS images.

Preprocessing techniques that we explore in this study are most closely related to synthesis strategies discussed above, but different from existing methods they do not try to synthesize images in the opposite domain. Instead they aim to minimize the difference in appearance by modifying images in both domains.

2.1 Deep neural networks for face recognition

Recently, face recognition has achieved significant progress thanks to the great success of DCNN-based approaches. The defining characteristic of such methods is the use of DCNNs as feature extractors. Implementations of DCNNs differ from each other mostly by architecture of DCNN and loss functions used for training. Taigman et al. [18], for example, presented DeepFace, which is a representative system of this class of methods. In DeepFace, face recognition is treated as a multi-class classification problem. The network's architecture consists of 9 layers and several locally connected layers. The model is trained with a K-way softmax with standard gradient descent. Schroff et al. [19] presented the FaceNet DCNN model, which directly learns an embedding into an Euclidean space for face verification. Parkhi et al. [10] presented a more systematic approach to face recognition using deep learning, with a semi-automatically collected dataset of millions of images and a principled model architecture consisting of 16 convolutional layers, deeper than was previously believes to be required for large-scale face recognition. More recently, He et al. [20] proposed a residual learning framework to train very deep networks (up to 152 layers). In recent studies, the problem of lack of power of discrimination in loss functions was addressed. Wen et al. [21] proposed the so-called center loss to learn centers for deep features of each identity and used the centers to reduce intra-class variance. Liu et al. [22] proposed a large margin softmax (L-Softmax) by adding angular constraints to each identity to improve feature discrimination. Deng et al. [23] propose an Additive Angular Margin Loss function to further improve the discriminative power of the face recognition model and to stabilize the training process.

3 Methodology

We now present the methodology used for your study including the selected preprocessing techniques and DCNN model considered.

3.1 Overview

Figure 2 shows an overview of the methodology used in this study. We use the Viola-Jones detector to detect faces in the input images and crop them to a standard size after detection. The cropped faces are then preprocessed with 9 different methods, which results in 10 sets of images for the experiments when also considering the original images. For every set of preprocessed images, feature extraction is performed using a DCNN model pretrained for recognition on VIS images. For verification experiment the Euclidean distance measure is used. An overview of the overall methodology is shown on Figure 2.

3.2 Preprocessing techniques

The aim of preprocessing the images is to reduce the visual differences between both modalities, while still preserving the intrinsic features so that the inter-subject variation is not compromised. In this study we consider 9 preprocessing techniques for this purpose and describe them in more detail below.

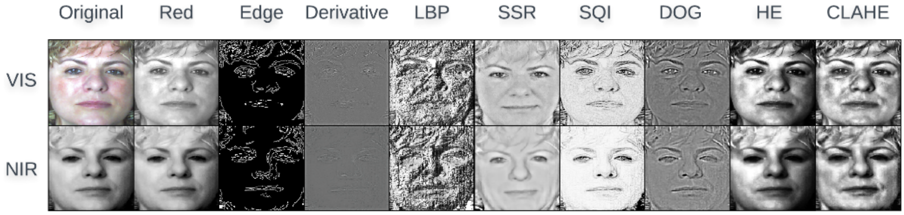


Figure 3: Output images of all preprocessing techniques on VIS and NIR

- **Red channel:** We extract the red channels from the VIS (RGB) image and compare it to the NIR images. The motivation behind this technique is to select the color channel closest in wavelength to infrared - the red channel corresponds to light wavelengths of *620-740nm*, whereas the near-infrared spectrum covers the wavelength range of *740-1400nm*.
- **Canny edge detector:** The Canny edge detection algorithm is applied to all the images, to extract edges of faces. The algorithm consists of 5 steps: noise reduction, gradient calculation, non-maximum suppression, double thresholding and edge tracking by hysteresis and is described in [24].
- **Laplacian:** The Laplacian of an image is a 2-D isotropic measure of the 2nd spatial derivative of an image. It highlights regions of rapid intensity change. The Laplacian $L_p(x, y)$ of an image with pixel intensity values $\mathrm{I(x, y)}$ is given by

$$L_p(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

and can in practice be calculated by a convolutional filter [25].

- **Local Binary Patterns (LBPs):** The LBP operator replaces the value of the pixels of an image with vectors of binary values, which are called LBP codes, that encode the local structure around each pixel [26] [27]. Each central pixel f_c is compared with its P neighbors $f_p$on radius R. The neighbors with a smaller value than the central pixel are assigned a value of 0, and the remaining neighbors are assigned a value of 1. For each central pixel, one can generate a binary number that is obtained by concatenating all these binary bits in a clockwise manner, starting in top-left neighbor. The resulting vector of binary numbers, interpreted as a decimal value of the generated binary number, replaces the central pixel value, and can be calculated by

–

$$LBP_{P,R}(x,y) = \sum_{p=0}^{P-1} s(f_p - f_c)2^P$$

$$s(v) = \begin{cases} 1; v \geq 0 \\ 0; v < 0 \end{cases}$$

– **Single Scale Retinex (SSR):** The SSR algorithm, proposed by Jobson et al. in [28], is based on the retinex theory. The theory models an image as a product of two components, i.e. illumination L and reflection component R. The key of SSR is to estimate the illumination of an image, and to restore the reflection component. The result of the SSR transformation is the reflection R, which can be computed by subtracting the estimated illumination $\hat{L}$ from the original image in the logarithm domain. The illumination is typically estimated by filtering the original image with a Gaussian filter G. Mathematically, SSR is described by

$$R(x,y) = \log\big(I(x,y)\big) - \log\left(\hat{L}(x,y)\right)$$

$$\hat{L}(x,y) = I(x,y) \otimes G(x,y)$$

– **Single scale self-quotient Image (SQI):** The SQI algorithm was introduced to the field of face recognition by Wang et al. in [29]. The technique exhibits similarities to the single scale retinex technique, but unlike the SSR technique uses an anisotropic filter for the smoothing operation. It has been proposed for synthesizing an illumination normalized image from the given face image. The SQI output Q is defined by image intensity value I and a smoothed image S, as

$$Q(x,y) = \frac{I(x,y)}{S(x,y)} = \frac{I(x,y)}{I(x,y) \otimes F(x,y)}$$

- **Difference of Gaussians (DOG):** The DOG technique relies on the difference of Gaussian filters to produce a normalized image. It effectively applies a band-pass filter to the input image and produces a normalized version. Before the filtering, gamma correction is applied. DOG filtering typically removes the-low frequency components of an image.
- **Histogram equalization (HE):** The HE algorithm [29] is a method of contrast adjustment using the image's histogram. It usually increases the global contrast of images. Through this adjustment, the intensities can be better distributed over the full dynamic range of the image. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. In our case it normalizes the images toward the same contrast distribution.
- **Contrast limited adaptive histogram equalization (CLAHE):** The CLAHE method [30] examines a histogram of intensities in a contextual region centered at each pixel and sets the displayed intensity at the pixel as the rank of that pixel's intensity in its histogram. CLAHE algorithm differs from standard HE in the respect that CLAHE operates on small regions in the image, called tiles, and computes several histograms, each corresponding to a distinct section of the image and uses these to redistribute the pixel values of the image.

Example results of the preprocessing with all considered preprocessing techniques are shown in Figure 3.

3.3 Off-the-shelf DCNN

For feature extraction, the VGGFace model [10] is used. It consists of a very deep CNN, in the sense that the network comprises a long sequence of convolutional layers, specifically 16 layers. Convolutional layers have small kernels and are followed by ReLU activations and max pooling layers. The model is pretrained on a very large dataset of \$2.6\$ million faces. A softmax activation function in the output layer is used for training, and a triplet loss function for fine-tuning. To extract a feature vector $\boldsymbol{x}$ from an input image with the VGGFace model, the softmax (classification) layer is discarded and activations of the last densely connected layers are used as the feature representation of the input face image.

3.4 Similarity comparison

For the similarity comparison of two feature vectors, extracted with the VGGFace model, the Euclidean distance is used. The Euclidean is the straight-line distance between two points in Euclidean space. If $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are feature vectors in Euclidean space extracted with VGGFace model the Euclidean distance is then defined as

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2.$$

4 Experiments

In this section the datasets, experimental protocol and metrics used for the evaluation are presented.

4.1 Datasets

For testing different preprocessing techniques, two datasets are used, CARL and NIVL. Example images from the datasets are shown in in Figure 4 and Figure 5, respectively.

CARL: The CARL dataset [31] [32] consists of a total of 7380 images in the visual spectrum, near-infrared (NIR) spectrum and thermal spectrum. It consists of 41 people, 32 males and 9 females. In each recording session the images were captured under three different illumination conditions. First natural illumination with sunlight entering through windows. Obviously, this illumination is not constant along days and a function of the different hours of the day. Second, infrared Illumination with circuit board around the webcam. And third artificial illumination with cool white fluorescence. All faces are frontally posed and are not tilted. There are also no occlusions. Each user was recorded in four different acquisition sessions over a period of one year. In this sense, there are distinctive changes in the haircut and/or facial hair in some of the subjects. Each individual contributed in four acquisition sessions and provided five different snapshots in three different illumination conditions and under three image sensors. For our experiments only images in the visual and near-infrared spectra are used, which means 60 images per person in one modality - in total, 4920 images.

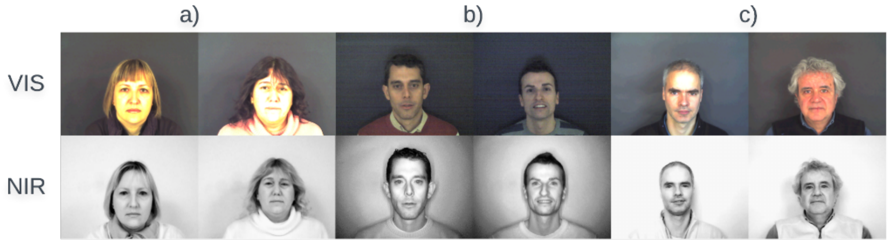


Figure 4: Example images from the CARL dataset. In first row presents VIS images and in second row presents NIR images. In each column image from the same person are shown. a) images taken with natural illumination with sunlight entering through windows, b) images taken under infrared illumination, and c) images taken under artificial VIS illumination.

NIVL: The NIVL dataset [33] contains 574 subjects. There are a total of 2341 VIS images and 22 264 NIR images from the 574 subjects. 402 subjects had both VIS and NIR images acquired during at least one session. Both VIS and NIR images were acquired in the same session, although not simultaneously. One VIS image and around 10 NIR images were acquired per subject per session. All faces are frontal and not tilted. Faces in NIR are taken close-up and consequently the top of the head and the bottom of the chin are cut off, while faces in VIS are visible completely. The NIR images have a resolution of 4770x3177 and the visible light images have a resolution of 4288x2848. Originally this dataset was designed and released by the University of Notre Dame with the intention of evaluate error rates of commercial face recognition matchers in the VIS-NIR task under different image processing algorithms. Since there is no need to train background models for commercial matchers, the original dataset evaluation protocol does not have a training set.

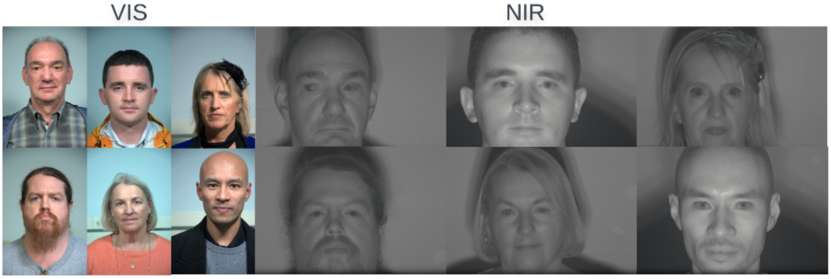


Figure 5: Example images from the NIVL dataset. In first three columns show VIS images and in last three columns show NIR images. For each of six people shown, there is one VIS and one NIR image.

4.2 Experimental setup

In most HFR problems, there is an image in NIR, called the probe image, that needs to be matched to one of the identities with corresponding images in the VIS dataset, called the gallery set. Accordingly, our datasets are divided into a gallery set of VIS images, and a probe set of NIR images.

The VGGFace model requires an 224x224 pixel image of a face at the input. Therefore, face detection is first applied using the Viola-Jones algorithm [34]. Faces are then cropped and resized to 224x224 pixels. All preprocessing is done on cropped and resized images of faces.
A verification experiment is carried out, where for every NIR image, a verification tests with all VIS images are conducted. Tests are of two types. Genuine attempts - when a pair consisting of a NIR image and a VIS image both belong to the same identity, and impostor attempts - when images in the pair belong to different identities. There are 147 600 genuine attempts and 5 904 000 impostor attempts for the CARL dataset, and 2 414 genuine attempts and 601 086 imposture attempts for the NIVL dataset in our experimental setup. Verification experiments are repeated for all preprocessing techniques.

4.3 Metrics

To evaluate the performance of HFR with different preprocessing techniques, the following metrics are used.

Results are graphically presented with Receiver Operating Characteristic (ROC) curves [35]. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. TPR measures the quantity of positives that are correctly identified as such. FPR measures the quantity of negatives that are wrongly identified as positive. On ROC random classification is presented with straight line between the points (0,0) and (1,1). Classification better than random is presented with line above the random line, the closer it is to point (0,1), the better the performance.

The quantitative measures that relate to ROC are the area under the ROC curve (AUC) and the equal error rate (EER). AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. For random classification, AUC equals 0.5, and for perfect classification, AUC equals 1. EER represents the value where 1-TPR equals FPR, i.e., the point where the false positive and false negative rates are equal. Smaller EER implies better verification performance.

4.4 Experimental results

Impact of preprocessing on HFR: Evaluation is done using verification experiment on both datasets separately. ROC curves for the verification experiments on both datasets are shown in Figure 6.

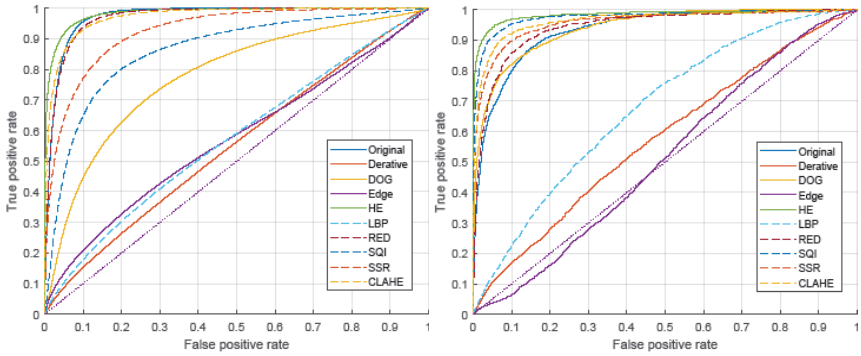


Figure 6: ROC results on the CARL (left) and NIVL (right) datasets.

Table 1: Quantitative metrics of our ROC experiments.

	CARL		**NIVL**		**CARL VIS-VIS**
	AUC	EER	AUC	EER	AUC
Original	97.4%	7.5%	93.5%	13.8%	99.7%
Red	97.1%	8.4%	94.4%	12.3%	99.5%
Edge	57.1%	45.0%	51.3%	49.3%	66.1%
Laplacian	55.0%	46.8%	57.6%	44.5%	62.1%
LBP	57.2%	45.1%	67.7%	37.7%	59.2%
SSR	92.7%	15.2%	95.8%	10.2%	99.5%
SQI	86.4%	19.9%	97.6%	7.0%	88.4%
DoG	77.8%	28.1%	94.1%	14.1%	83.3%
HE	98.5%	6.4%	98.4%	5.5%	99.9%
CLAHE	97.3%	8.5%	96.6%	8.7%	99.2%

AUC and EER values to corresponding ROC curves are listed in Table 1. ROC curves that belong to verification experiments on CARL dataset using only VIS images and VIS-VIS matching are shown on Figure 7.

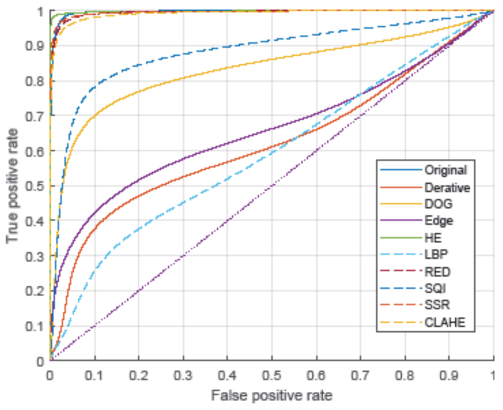


Figure 7: ROC results for the CARL dataset VIS-VIS matching experiment

As can be seen, the preprocessing techniques that transform input images pixels to binary values, i.e., the LBP transformation and Edge detection show the poorest results. This is likely because the VIS-trained DCNN networks expect natural-looking images as inputs as opposed to binarized ones. The preprocessing with Laplacian image derivative also shows uncompetitive results. With all those techniques verification performance drops in comparison to no preprocessing on both datasets % their AUC values show barely above random recognition.

Normalizing images using the DOG technique also decreases verification performance on the CARL dataset. The EER value of this technique on the NIVL dataset is also higher, which means lower performance than on the original images. However, the AUC value is slightly higher pointing to a somewhat better performance overall. The CLAHE, SSR, SQI and red channel techniques show an increase in recognition performance on the NIVL dataset, and a slight decrease in performance on the CARL dataset. The success of preprocessing techniques varies considerably from dataset to dataset. For example, the SQI method gives much better results on NIVL dataset than on CARL dataset. The most successful preprocessing technique, with the highest AUC value and the lowest EER value on both datasets is histogram equalization, which appears to be beneficial in bringing the visual appearance of images of both domains closer together, but also produces images well suited for the VGG model.

Impact of preprocessing on VIS: The differences in verification performance due to the preprocessing technique is not only a consequence of the decrease of the appearance gap between modalities, but also because the VGG model is not trained on images preprocessed with such techniques. Another verification experiment was, therefore, performed on the CARL dataset with only VIS images and VIS to VIS matching to explore the effect of different preprocessing techniques on recognition of VIS images with the same DCNN model. ROC curves of this experiments are shown in Figure 7. The comparison of all experiments on the CARL dataset can be seen more clearly on Figure 8 where AUC values of all preprocessing techniques and with both NIR-VIS and VIS-VIS matching are presented on a bar chart.

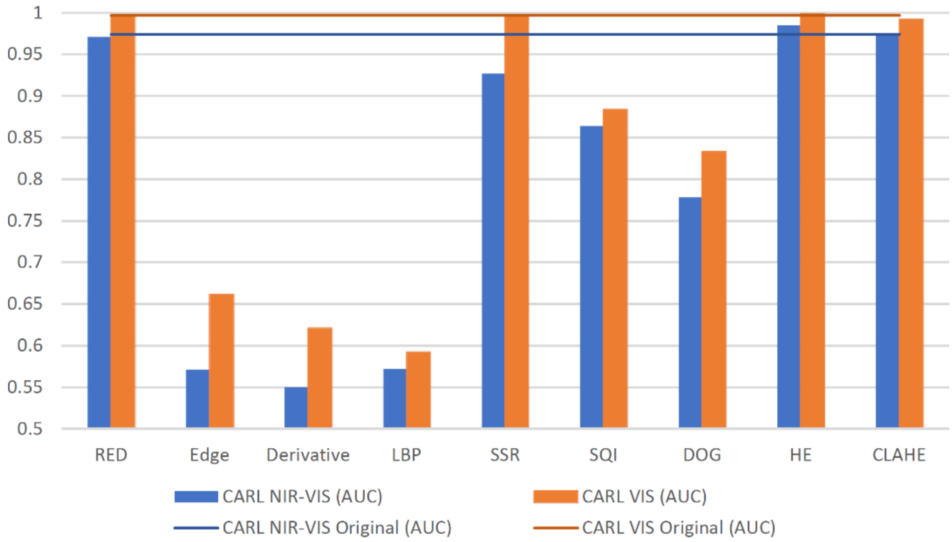


Figure 8: AUC values of ROC curves that correspond to experiments on the CARL dataset. The blue bars show AUC values of NIR-VIS matching for every preprocessing technique, and the blue line presents AUC value for original images. The red bars show AUC values of VIS-VIS matching for every preprocessing technique, and the red line presents AUC value for original images.

5 Conclusions

In this paper we investigated how different preprocessing techniques applied on VIS and NIR images affect HFR performance with DCNN pretrained for recognition of VIS images. VGGFace was used for feature extraction. Performance was tested on two datasets, CARL and NIVL. The preprocessing techniques were applied on all images in both VIS and NIR spectra. After feature extraction, a verification experiment with all VIS - NIR couples was carried out. We observed that performance of such systems can increase only by comparing NIR images to red channel of VIS. The best results of all tested preprocessing techniques was achieved with the HE method, which increased performance of HFR on both datasets.

References

[1] X. Wang and X. Tang, "Face Photo-Sketch Synthesis and Recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, pp. 1955-67, 11 2009.

[2] Z. Deng, X. Peng, Z. Li and Y. Qiao, "Mutual Component Convolutional Neural Networks for Heterogeneous Face Recognition," *IEEE Transactions on Image Processing*, vol. PP, pp. 1-1, 1 2019.

[3] Y. Jin, J. Lu and Q. Ruan, "Coupled Discriminative Feature Learning for Heterogeneous Face Recognition," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 640-652, 3 2015.

[4] S. Wang, L. Zhang, Y. Liang and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in 2012 *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[5] G. Hu, X. Peng, Y. Yang, T. Hospedales and J. Verbeek, "Frankenstein: Learning Deep Face Representations Using Small Data," *IEEE Transactions on Image Processing*, vol. PP, 3 2016.

[6] X. Tang and X. Wang, "Face Sketch Recognition," *Circuits and Systems for Video Technology, IEEE Transactions*, vol. 14, pp. 50-57, 2 2004.

[7] J. Zhong, X. Gao and C. Tian, "Face Sketch Synthesis using E-HMM and Selective Ensemble," 2007.

[8] D. Lin and X. Tang, "Inter-modality Face Recognition," 2006.

[9] S. Liu, D. Yi, Z. Lei and S. Li, "Heterogeneous face image matching using multi-scale features," 2012.

[10] O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep Face Recognition," in *British Machine Vision Conference*, 2015.

[11] S. Ouyang, T. M. Hospedales, Y. Song, X. Li, C. C. Loy and X. Wang, "A survey on heterogeneous face recognition: Sketch, infra-red, 3D and low-resolution," *Image Vision Comput.*, vol. 56, pp. 28-48, 2014.

[12] D. Gong, Z. Li, W. Huang, X. Li and D. Tao, "Heterogeneous Face Recognition: A Common Encoding Feature Discriminant Approach," *IEEE Transactions on Image Processing*, vol. PP, pp. 1-1, 1 2017.

[13] K. Wang, R. He, W. Wang, L. Wang and T. Tan, "Learning Coupled Feature Spaces for Cross-Modal Matching," 2013.

[14] B. Riggan, C. Reale and N. M. Nasrabadi, "Coupled Auto-Associative Neural Networks for Heterogeneous Face Recognition," *Access, IEEE*, vol. 3, pp. 1620-1632, 1 2015.

[15] Y. Zhuang, Y. Wang, F. Wu, Y. Zhang and W. Lu, "Supervised coupled dictionary learning with group structures for multi-modal retrieval," *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, AAAI 2013, pp. 1070-1076, 1 2013.

[16] T. Freitas Pereira, A. Anjos and S. Marcel, "Heterogeneous Face Recognition Using Domain Specific Units," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 1803-1816, 7 2019.

[17] Q. Liu, X. Tang, H. Jin, H. Lu and S. Ma, "A nonlinear approach for face sketch synthesis and recognition," 2005.

[18] T. Yaniv, Y. Ming and W. Lior, "Deepface: Closing the gap to human-level performance in face verification," in In: *IEEE CVPR*, 2014.

[19] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015.

[20] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016.

[21] Y. Wen, K. Zhang, Z. Li and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," 2016.

[22] W. Liu, Y. Wen, Z. Yu and M. Yang, "Large-Margin Softmax Loss for Convolutional Neural Networks," *ProC. Int. Conf. Mach. Learn.*, 12 2016.

[23] J. Deng, J. Guo and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," 1 2018.

[24] J. Canny, "A Computational Approach To Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions*, Vols. PAMI-8, pp. 679-698, 12 1986.

[25] R. M. Haralick and L. G. Shapiro, Computer and robot vision, vol. 1, Addison-wesley Reading, 1992.

[26] T. Ojala, M. Pietikainen and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proceedings of 12th International Conference on Pattern Recognition*, 1994.

[27] T. Ojala, M. Pietikäinen and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, pp. 51-59, 1996.

[28] D. J. Jobson, Z. Rahman and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE Transactions on Image Processing*, vol. 6, pp. 451-462, 3 1997.

[29] Y. C. Hum, K. W. Lai and M. I. Mohamad Salim, "Multiobjectives Bihistogram Equalization for Image Contrast Enhancement," *Complex.*, vol. 20, p. 22–36, 10 2014.

[30] S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas and K. E. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," in [1990] *Proceedings of the First Conference on Visualization in Biomedical Computing*, 1990.

[31] V. Espinosa-Duró, M. Faundez-Zanuy and J. Mekyska, "A New Face Database Simultaneously Acquired in Visible, Near-Infrared and Thermal Spectrums," *Cognitive Computation*, vol. 5, pp. 119-135, 3 2013.

[32] V. Espinosa-Duró, M. Faundez-Zanuy, J. Mekyska and E. Monte-Moreno, "A Criterion for Analysis of Different Sensor Combinations with an Application to Face Biometrics," *Cognitive Computation*, vol. 2, pp. 135-141, 9 2010.

[33] J. Bernhard, J. Barr, K. W. Bowyer and P. Flynn, "Near-IR to visible light face matching: Effectiveness of pre-processing options for commercial matchers," in 2015 IEEE *7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2015.

[34] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001.

[35] T. Fawcett, "Introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861-874, 6 2006.

PUPILLARY DISTANCE MEASUREMENT WITH FULLY CONVOLUTIONAL NEURAL NETWORK

NIKO GAMULIN & BLAŽ MEDEN

University of Ljubljana, Faculty of Computer and Information Science, Computer Vision Laboratory, Ljubljana, Slovenia, e-mail: niko.gamulin@fri.uni-lj.si, blaz.meden@fri.uni-lj.si

Abstract In this paper we present the practical application of numerical coordinate regression with convolutional neural network for pupillary distance measurement. In case of applying deep learning to a specific domain, it is difficult to construct a large-scale dataset, which limits the end-to-end learning potential of complex network architectures. Therefore, the problem has to be addressed either by applying transfer learning or design an architecture with a complexity level much lower than state of the art models. For pupillary distance measurement, the latter approach has led to satisfactory results.

Keywords: convolutional neural networks, stacked hourglass, numerical coordinate regression, pupillary distance, heatmap.

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-337-1.3
ISBN 978-961-286-337-1

1 Introduction

Pupillary distance (PD) is the distance measured in millimiters between the centers of the pupils of the eyes. For people who need to wear prescription glasses, PD measurement, usually measured by the optician helps to ensure that the lenses are located in the optimum position. Prior to image processing-based approaches, pupillometer [1] has been patented and it has been widely adopted up to date. In recent years, deep convolutional neural networks (CNNs) have proven to be highly effective general models for various computer vision problems [2]–[4]. One such problem is coordinate regression, where the goal is to predict a fixed number of location coordinates, corresponding to points of interest in an input image.

By detecting the edge coordinates of an object of known size, it is possible to determine the pixel per unit of size ratio. Consequently, it is possible to calculate the absolute distance between the arbitrary coordinates pair on an image of interest.

To achieve satisfactory precision of the measurements, however, it is necessary to construct large enough dataset that contains images with both size reference object and landmarks between which the distance has to be measured.

In case of having a small dataset, the available options to address the limitation is either to apply transfer learning [5] or define the network architecture with lower complexity, compared to the complexities of the most commonly used backbones, such as VGG [6], ResNet [7], and InceptionNet [8].

For PD measurement, we used a standard card of dimensions 85.60 × 53.98 mm; the most common shape of credit cards, personal identification cards, and loyalty cards. The number of images collected for this task was relatively small, and therefore, we first applied transfer learning and later constructed a fully-convolutional neural network [3] with a small number of layers. For transfer learning, we tried to use several VGG, ResNet and InceptionNet architecture, pre-trained on ImageNet [9] and fine-tune custom head. As none of the transfer learning attempts provided promising results, we used fully convolutional network for semantic segmentation with output transformation.

2 Related work

In [10], Sun et.al. applies a three-level convolutional neural network to obtain landmark estimation. In [4], Newell et. al. proposed a stacked hourglass network architecture to capture various spatial relationships associated with the body. In [11], Yang et. al proposed stacked hourglass architecture for facial landmark localisation. In [12], Nibali et.al. proposes differentiable spatial to numerical transform to calculate landmark coordinates from heatmaps.

3 Pupillary distance measurement

In this chapter, we present a network architecture and the transformations, applied to coordinate labels to get the heatmaps and network output to calculate coordinates from heatmaps.

3.1 Network architecture

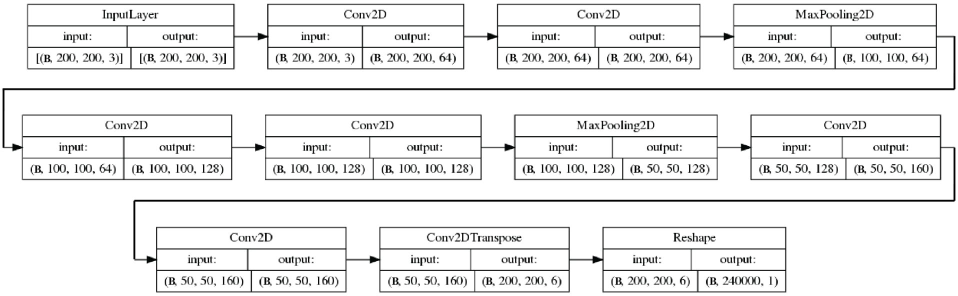


Figure 1: Our Fully Convolutional Network

Similar to human pose estimation [1], we implemented a stacked hourglass network [4] to detect card corner and pupil landmarks. Input images have been resized to 200 x 200px and passed in the batch of size B through several convolutions, max pooling and deconvolution layers as displayed on Figure 1. The output is a 6-channel layer that represents 200 x 200px masks for left and right pupils and top-left, top-right, bottom-left, and bottom-right card corner. The transformation from masks to points is described in section 3.3.

3.2 Coordinates to heatmaps transformation

On the images there are at most six landmarks visible: pupils and card corners. Therefore, every image has been initially labelled with a 12-dimensional vector that contains x and y value for every landmark. In case the landmark is not visible, the x and y value for the expected landmark has been set to null.

For every potentially visible landmark, a black image (zero values) array of size 200 x 200 px has been created. After, for every visible landmark, a Gaussian kernel with a selected standard deviation □ around the x and y coordinate on the black image has been added according to equation (1).

$$f(x,y) = \frac{(x - L_x)^2 + \left(y - L_y\right)^2}{2\sigma^2} \tag{1}$$

3.3 Heatmaps to coordinates transformation

In order to determine the predicted coordinates, each of 6 heatmaps, representing a potential landmark, is processed as follows. The pixel coordinates are sorted by brightness h in descending order. Then, a subset of top n brightest coordinates is selected to calculate the mean coordinate values and mean brightness. If the calculated brightness value is above the defined threshold, it is assumed that the heatmap contains a valid landmark.

$$x = \frac{\sum_{i=1}^{n} x_i h_i}{\sum_{i=1}^{n} h_i} \tag{2}$$

$$y = \frac{\sum_{i=1}^{n} y_i h_i}{\sum_{i=1}^{n} h_i} \tag{3}$$

The calculation procedure is described in a simplified example. Figure 2 represent a heatmap of size 6 x 6 px with values in range (0, 255). In case of selecting top 5 brightest pixels, the following array indices (1, 5), (3, 2), (5, 3), and (3, 3) are selected with belonging values 205, 206, 212, 218, 233. According to equations (2), (3), the calculated x and y coordinate values are 2.6 and 2.19, respectively, and the calculated brightness is 214.8.

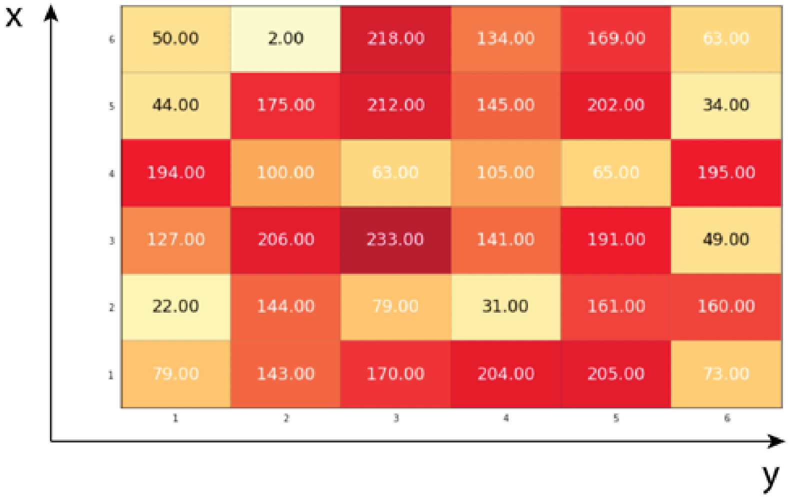


Figure 2: Heatmap example

3.4 Loss function

As the output heatmaps are transformed to coordinates, it is possible to calculate directly two-dimensional Euclidean distance between the prediction $\hat{y}$ and ground truth y. According to this fact, the loss function is formulated as (4).

$$L(\hat{y}, y) = \|y - \hat{y}\|_2 \tag{4}$$

4 Results

4.1 Dataset

To train and evaluate the model, we have collected 560 images. Major percentage of the images were collected by peers who were instructed to hold the card in front of the mouth. Additionally, we have added a smaller number of images collected from the internet. that contain a person's face and a card. Each image has been annotated with a 12-dimensional vector that represents 6 landmarks: left (LP), right (RP) pupil, top-left (CTL), top-right (CTR), bottom-right (CBR), and bottom-left (CBL) card corner.

x	y	x	y	x	y	x	y	x	y	x	y
LP		RP		CTL		CTR		CBR		CBL	

Figure 3: Original label representing a vector of x and y coordinated for left pupil (LP), right pupil (RP), top-left (CTL), top-right (CTR), bottom-right (CBR) and bottom-left (CBL) card corner.

In order to artificially increase the number of images, the dataset has been augmented by applying affine transformations (padding, rotation, translation, horizontal flipping, brightness and contrast adjustment). 80% of images have been used for training and the remaining 20% for evaluation.

4.2 Qualitative results

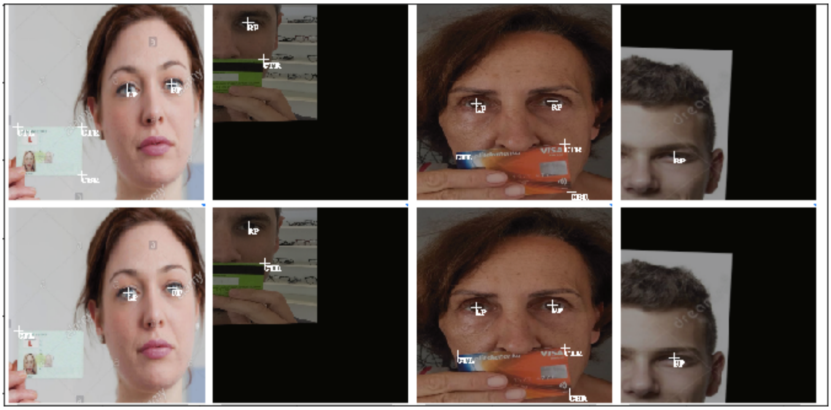


Figure 4: Comparing predictions (bottom row) with ground truth (top row)

Figure 4 shows the comparison of predicted landmarks in the bottom row with ground truth in the top row. Due to image augmentations, the images are randomly padded, rotated, cropped, and translated. In case of correct predictions (second, third, and fourth image from the left), the distance between the predicted and ground truth landmark is negligible. In some instances, however, the landmarks are not detected (first image from the left). As the position of the card relative to pupils in the dataset doesn't vary significanly (instances of peers holding the card in front of the mouth), one possible reason for detection failure might be a lack of variations in the training set and consequently overfitting of the model to expected card position.

4.3 Loss convergence

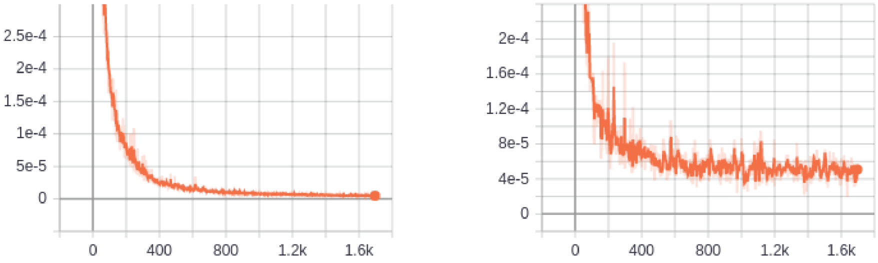


Figure 5: Training (left) and test (right) loss values (y) axis throughout epochs (x) axis.

Figure 5 shows the loss values according to equation (4) for normalized values y and $\hat{y}$ throughout the training of 1800 epochs. The batch size has been set to 64 and the training on NVIDIA RTX 2080Ti took approximately 12 hours. By comparing the training and test loss, we assumed that the trained model does not overfit.

5 Conclusion

In this paper, we proposed a fully convolutional neural network for PD measurement. We demonstrated how output heatmaps are transformed to coordinate values, which are compared to ground truth values.

There are several hyperparameters to be explored which impact the model performance. Currently, we just implemented a fully convolutional network and trained it from scratch. In the future, we plan to examine additional models with different number of layers and perform quantitative analysis. Additionally, we plan to determine the reasons for detection failures and improve the robustness of the model in order to correctly detect the landmarks for arbitrary card position relative to pupils.

References

[1] A. C. Macbeth, "Instrument for measuring pupillary distances," 2,596,264, 1952.

[2] A. Krizhevsky and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., pp. 1097–1105, 2012.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," Proc. IEEE Conf. Comput. Vis. pattern Recognit., pp. 3431–3440, 2015.

[4] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," Eur. Conf. Comput. Vis., pp. 483--499, 2016.

[5] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," arXiv Prepr. arXiv1911.02685, no. 10, pp. 1–27, 2019.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv Prepr. arXiv1409.1556, pp. 1–14, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[8] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in CVPR09, 2009.

[10] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 3476–3483.

[11] J. Yang, Q. Liu, and K. Zhang, "Stacked hourglass network for robust facial landmark localisation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 79–87.

[12] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "Numerical coordinate regression with convolutional neural networks," arXiv Prepr. arXiv1801.07372, 2018.

VIZUALIZACIJA VZORCA SOŽARIŠČNE MIKROSKOPIJE S TEHNOLOGIJO OBOGATENE RESNIČNOSTI

KLEMEN BABUDER[1], LUKA GANTAR[2], BORUT BATAGELJ[1] IN FRANC SOLINA[1]

[1] Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija, e-pošta: klemen.babuder@ltfe.org, borut.batagelj@fri.uni-lj.si, franc.solina@fri.uni-lj.si
[2] Univerza v Manchesteru, Manchester, Združeno kraljestvo Velike Britanije in Severne Irske, e-pošta: luka.gantar@student.manchester.ac.uk

Povzetek V okviru tega prispevka je opisana aplikacija za pametna očala Microsoft HoloLens. Aplikacija uporabniku omogoča ogled in interakcijo s tridimenzionalnim vzorcem s pomočjo tehnologije obogatene resničnosti. Uporabljen je vzorec živčnega tkiva gensko spremenjene miši pridobljen s sožariščno mikroskopijo. Skozi holografski vmesnik lahko uporabnik spreminja lastnosti vzorca in pri tem izlušči dodatne informacije.

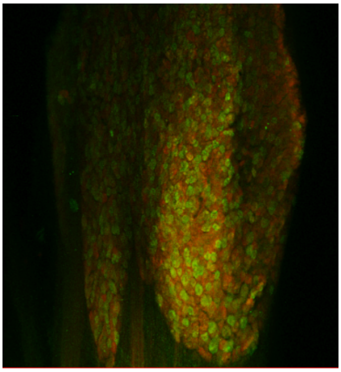
Ključne besede:
obogatena resničnost,
Microsoft HoloLens,
sožariščna mikroskopija,
napredna vizualizacija,
holografija.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1.5
ISBN 978-961-286-337-1

1 Uvod

Sožariščna mikroskopija je mikroskopska tehnika, ki se od klasične (presevne) mikroskopije razlikuje v tem, da naenkrat osvetli le eno rezino vzorca. Z združevanjem rezin na različnih globinah ustvari kopico slik, ki se jih lahko pretvori v trirazsežne modele. Z dodajanjem fluorescentnih označevalcev se lahko dodatno izrazi podrobnosti bioloških vzorcev, ki s klasično mikroskopijo niso vidne [1].



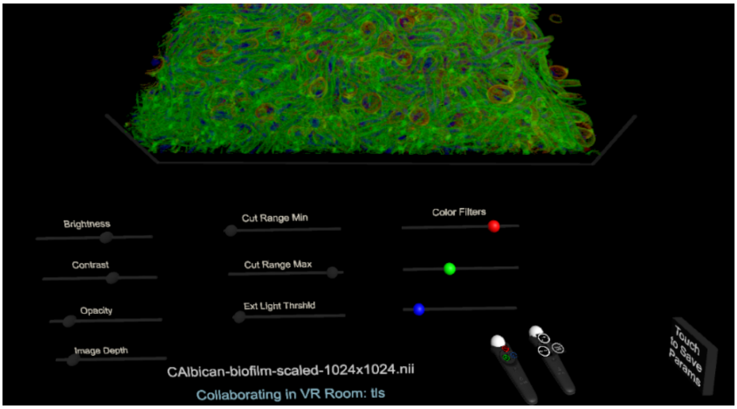
Slika 1: Tridimenzionalna projekcija vzorca živčnega tkiva z obarvanim GFP (zelena barva) in nevronskimi jedri (rdeča barva) v programu Fiji.

Za ogled vzorcev sožariščne mikroskopije znanstveniki pogosto uporabljajo program Fiji ali Fiji is Just ImageJ, ki predstavlja distribucijsko različico programa ImageJ. ImageJ je odprto kodni program namenjen analizi znanstvenih slik [2]. Fiji v okviru svojih funkcionalnosti uporabniku ponuja ogled posameznega sloja vzorca ter ogled celotne kopice vzorcev v obliki tridimenzionalnega modela.

Fiji in njemu podobna programska oprema omogoča izris in ogled tridimenzionalnih slik na dvodimenzionalnih zaslonih, kjer so uporabniki primorani rotirati sliko, da odkrijejo celotno tridimenzionalno strukturo. S tehnologijo navidezne in izboljšane resničnosti se znanstvenikom ponuja boljši način ogleda tovrstnih struktur, kjer se vzorce prikazuje v trirazsežnem prostoru, ki je človeku bolj naraven in intuitiven.

Podjetje Immersive Science LLC je v preteklem letu predstavilo aplikacijo ConfocalVR, ki omogoča ogled vzorcev sožariščne mikroskopije v okolju navidezne resničnosti (angl. virtual reality). Aplikacija je bila predstavljena v okviru strokovno pregledanega članka z naslovom »ConfocalVR: Immersive Visualization for Confocal Microscopy« [3].

Ogled in interakcija z vzorci v trirazsežnem prostoru navidezne realnosti pomaga znanstvenikom priti do novih odkritji na področju zdravstva in dodaja znanosti novo dimenzijo [4, 5].



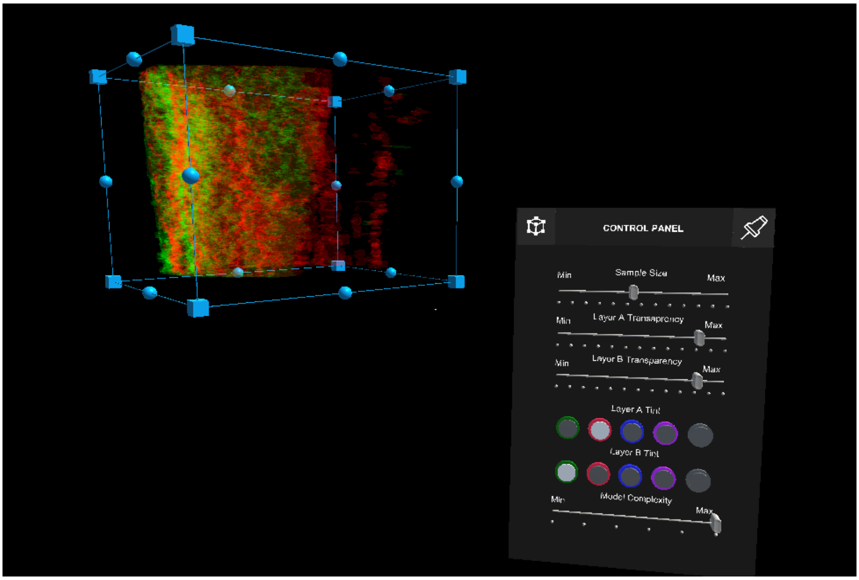
Slika 2: Slika prikazuje navidezni model sožariščnega vzorca in uporabniški vmesnik v aplikaciji [6].

2 Aplikacija za pametna očala Microsoft HoloLens

Nadgradnja izkušnje ogledovanja vzorcev sožariščne mikroskopije iz tradicionalnega načina na ogledovanje v navidezni resničnosti prinaša znanosti veliko koristi. Smiselno je predpostaviti, da bi podobno, nadgradnja na ogledovanje v obogateni resničnosti tudi koristila znanosti. Poleg tega, pa bi tehnologija zaradi svojih sposobnosti prelivanja resničnega in navideznega sveta prispevala še nekoliko drugačen pogled ali celo dodala povsem novo dimenzijo.

2.1 Izdelana rešitev

Na sliki 3 je predstavljen prikaz vzorca in vmesnika, ki se ob zagonu aplikacije pojavita v prostoru. Če vzorec kadarkoli zapusti uporabnikovo vidno polje, se uporabniku pojavi rdeča puščica, ki kaže v smeri modela. S tem je uporabniku olajšano lociranje modela.



Slika 3: Vzorec in vmesnik prikazana v simuliranem okolju urejevalnika Unity.

Uporabniku je na desni strani vedno na voljo uporabniški vmesnik poimenovan nadzorna plošča (angl. Control Panel). Vmesnik uporabniku neprestano sledi po prostoru in mu tako ostaja na dosegu roke. Ozadje vmesnika sestavljajo temne barve, saj se hologrami gumbov, drsnikov in ostalih elementov vmesnika v prostoru najbolje opazijo, če so postavljeni pred črnim ozadjem.

V glavi uporabniškega vmesnika se nahajata dva gumba in ime vmesnika. Prvi omogoča izklop ali vklop mejne škatle in z njo možnosti premikanja objekta po prostoru. Drugi uporabniku omogoča, da vmesnik pripne na mesto, saj lahko neprestano sledenje vmesnika uporabnika ovira pri ogledovanju vzorca.

V telesu uporabniškega vmesnika se nahajajo štirje drsniki in dva seta gumbov. Prvi drsnik omogoča natančno prilagajanje velikosti vzorca. Drugi in tretji dresnik omogočata spreminjanje prosojnosti posameznega kanala vzorca. S tem uporabniku omogoča, da bolj izpostavi en ali drugi kanal. Sledita dva seta gumbov, ki sta namenjena spreminjanju barve posameznega kanala vzorca. S spreminjanjem barve vzorca se lahko v vzorcu bolje izrazijo podrobnosti posameznega kanala, z klikom na najbolj desni gumb, pa se lahko katerega koli od kanalov tudi povsem izklopi.

Zadnji drsnik omogoča uporabniku prilagajanje števila poligonov, ki izrisujejo model. Glede na izbrano vrednost drsnika se v aplikaciji model izrisuje z različnim številom trikotnikov. Višje število trikotnikov pomeni večjo kompleksnost modela ter posledično večji nivo podrobnosti in večjo procesorsko obremenitev za napravo.

3 Zaključek

V prispevku je opisana rešitev, ki omogoča vizualizacijo vzorca sožariščne mikroskopije s tehnologijo obogatene resničnosti na očalih Microsoft HoloLens. Uporaba aplikacije ni omejena samo na sožariščno mikroskopijo, saj omogoča prikaz vzorcev katerekoli tehnike, ki je sposobna generirati tridimenzionalne vzorce. Aplikacija se trenutno zanaša na različne programe za pretvorbo vzorcev v primerno obliko za ogled v obogateni resničnosti. Smiselna nadgradnja bi bila torej podporna storitev, ki bi uporabniku omogočala samodejno pretvorbo in uvoz vzorcev zajetih s sožariščno mikroskopijo v aplikacijo.

Literatura

[1] James B. Pawley (1995), Handbook of Biological Confocal Microscopy, Plenum Press.

[2] Fiji is just ImageJ, https://imagej.net/Fiji, Dostopano 08. 02. 2020.

[3] ConfocalVR: Immersive Visualization for Confocal Microscopy, https://www.ncbi.nlm.nih.gov/pubmed/29949752. Dostopano 10. 01. 2020.

[4] How virtual reality is helping scientists make new discoveries about our health, https://www.geekwire.com/2017/virtual-reality-helping-scientists-make-new-discoveries-health/, Dostopano 08. 02. 2020.

[5] Virtual-reality applications give science a new dimension, https://www.nature.com/articles/d41586-018-04997-2, Dostopano, 08. 02. 2020.

[6] Slika aplikacije Confocal VR, https://www.immsci.com/wp-content/uploads/2018/12/ConfocalVR_Example_Image2.png. Dostopano, 08. 02. 2020.

PREPOZNAVANJE AKTIVNOSTI OSEBE IZ ZAPOREDJA SLIK Z GLOBOKIMI POVRATNIMI NEVRONSKIMI MREŽAMI

DAVID PINTARIČ, MARTIN ŠAVC IN BOŽIDAR POTOČNIK

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Maribor, Slovenija, e-pošta:
david.pintaric@gmail.com, martin.savc@um.si, bozidar.potocnik@um.si

Povzetek V tem članku, v katerem povzemamo glavne rezultate diplomskega dela prvega avtorja, se ukvarjamo s problemom prepoznavanja aktivnosti osebe iz zaporedja slik. Prepoznavo želimo izboljšati z upoštevanjem časovne komponente. To dosežemo z uporabo povratnih nevronskih mrež. Omejili smo se na naslednje aktivnosti: oseba ni v ravnovesju, se pripogiba, stoji, sedi, leži, hitro in počasi hodi ter pada. Rezultati na 25 označenih videoposnetkih so pri uporabi povratne nevronske mreže pokazali 83,2 % povprečno natančnost pri uporabi tipa zaporedje v vektor in 75,5 % povprečno natančnost pri uporabi tipa zaporedje v zaporedje. Kljub temu da so dobljeni rezultati boljši od tistih, kjer ne upoštevamo časovne komponente, ugotavljamo, da povratne nevronske mreže zaradi računske zahtevnosti niso vedno najboljša izbira.

Ključne besede: računalniški vid, globoko učenje, pomnilna celica GRU, detekcija oseb, prepoznavanje aktivnosti osebe.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1.6
ISBN 978-961-286-337-1

1 Uvod

Področje prepoznavanja aktivnosti osebe je med raziskovalci vzbudilo interes že v 80. letih prejšnjega stoletja, saj ima veliko uporabnost na številnih drugih področjih, kot so medicina, interakcija človek-računalnik, nadzor in sociologija. Kljub številnim naporom in izjemnim uspehom trenutno še vedno ne dosega visokih standardov natančnosti. Prav tako je pravilna realizacija tega področja v nekaterih nalogah, kot je na primer videonadzor, še vedno odprt raziskovalni problem [1].

V tej raziskavi smo za reševanje problema prepoznavanja aktivnosti osebe iz zaporedja slik uporabili povratne nevronske mreže. Omejili smo se na prepoznavanje naslednjih aktivnosti: oseba ni v ravnovesju, se pripogiba, stoji, sedi, leži, hitro hodi, počasi hodi in pada. Omejeni smo bili z majhno učno množico, ki je vključevala le 25 različnih video posnetkov, na katerih so osebe izvajale prej naštete aktivnosti.

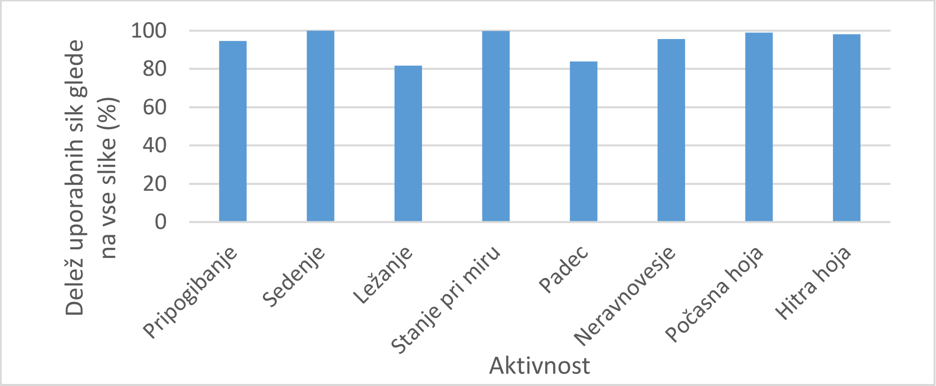
2 Algoritem

Algoritem, ki smo ga uporabili za prepoznavanje aktivnosti osebe iz zaporedja slik je bil sestavljen iz petih korakov, in sicer iz i) izločevanja oseb iz slik, ii) generiranja zaporedij, iii) obogatitve slik, iv) izločevanja značilnic iz posameznih slik in v) prepoznavanja aktivnosti osebe iz zaporedja slik.

2.1 Izločevanje osebe

Iz vsake slike v vsakem videoposnetku smo izločili osebo, ki se je na njem nahajala. Za izvedbo tega koraka smo uporabili predhodno naučen model ssd_resnet_50_fpn_coco iz Tensorflow-ove zbirke modelov za detekcijo objektov [2], ki smo ga izbrali na podlagi dobrega razmerja med natančnostjo in hitrostjo. Model na vseh slikah ni zaznal osebe. Prav tako smo ročno pregledali vse preostale slike in odstranili tiste, ki so prikazovale manj kot 75 % osebe.

Rezultati so pokazali, da je bila natančnost zaznave manjša od 90 % le pri dveh razredih, tj. ležanje in padec. Na sliki 1 je prikazan graf deleža uporabnih slik glede na vse slike pri izločanju oseb.



Slika 1: Delež uporabnih slik glede na vse slike pri izločanju oseb

2.2 Generiranje zaporedij

Za prepoznavanje aktivnosti oseb smo uporabili zaporedja slik. Pridobljene oklepajoče slike oseb smo zato združili v zaporedja. Zaporedja smo pridobili s postopkom drsečega okna velikosti 90, ki smo ga na začetku položili na prvih 30 slik in ga nato premikali po 10 slik naprej. S postopkom smo prenehali, ko je bilo pod drsečim oknom manj kot 30 sličic.

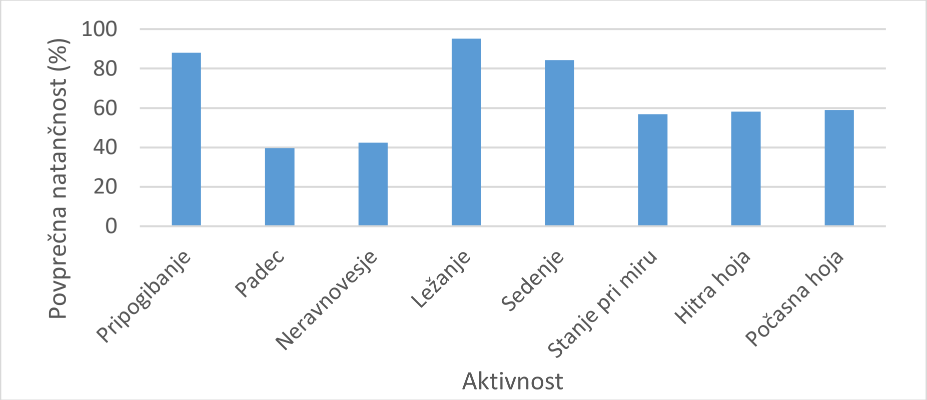
2.3 Obogatitev slik

Slike v naši podatkovni zbirki smo tudi obogatili. S tem smo umetno povečali velikost naše podatkovne zbirke, pri čemer smo obogatili le slike v učni množici. Uporabili smo naslednje transformacije: zasuk okoli osi y, gaussova zameglitev, normalizacija kontrasta, aditiven gaussov šum, posvetlitev in potemnitev slike, skaliranje po x in y osi, translacija in rotacija.

2.4 Izločevanje značilnic iz posameznih slik

Značilnice bi lahko izločili direktno z uporabo povratnih nevronskih mrež, vendar te niso uspešne pri procesiranju prostorskih informacij, ki se nahajajo v slikah. Odločili smo se za vmesen korak izločevanja značilnic z uporabo konvolucijske nevronske mreže. Uporabili smo predhodno naučen model imenovan Inception v3 [3], kateremu smo odstranili zadnje tri plasti.

Za lažjo in verodostojnejšo primerjavo uspešnosti različnih detekcijskih metod smo modelu Inception v3 za zadnjo povprečno združevalno plastjo dodali plast Softmax in jo učili na naši učni množici. Pri učenju smo zamrznili vse ostale plasti modela. Na sliki 2 vidimo graf povprečne natančnosti prepoznave posamezne aktivnosti po učenju zadnje plasti Softmax modela Inception v3, in sicer na osnovi zgolj ene slike. Končna povprečna natančnost našega modela za prepoznavo aktivnosti na osnovi zgolj ene slike je bila tako 69,4 %.



Slika 2: Povprečne natančnosti prepoznave posamezne aktivnosti osebe po učenju zadnje plasti Softmax modela Inception v3 na osnovi zgolj ene slike

2.5 Prepoznavanje aktivnosti iz zaporedja slik

Preizkusili smo dva tipa povratnih nevronskih mrež, in sicer i) tip zaporedje v vektor, ki vrne eno zaznano aktivnost za vsako zaporedje in ii) tip zaporedje v zaporedje, ki vrne zaznano aktivnost za vsako sliko v zaporedju. Zaradi hitrosti učenja in primerljivih rezultatov glede na pomnilno celico LSTM, smo se odločili za uporabo pomnilne celice GRU [4]. Vhod v nevronsko mrežo so sestavljala zaporedja, ki so vsebovala izhod zadnje povprečne združevalne plasti originalnega modela Inception v3.

3 Rezultati

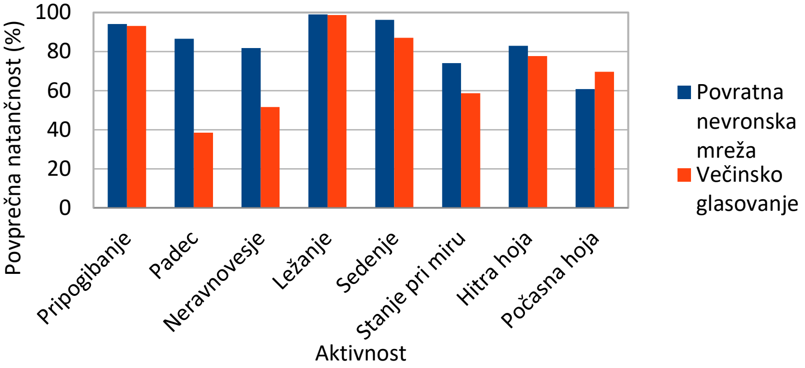
V nadaljevanju predstavljamo dobljene rezultate. Eksperimente smo izvajali na podatkovni zbirki, ki je bila sestavljena iz 25 že označenih videoposnetkov, ki so prikazovali različne osebe pri izvajanju različnih aktivnosti. Ozadje se med videoposnetki ni bistveno spreminjalo in na vsakem videoposnetku je bila prisotna le ena oseba hkrati. Videoposnetki so vsebovali sledeče aktivnosti, kjer je v oklepajih zapisano število slik, ki so vsebovale dano aktivnost:

- pripogibanje (14815)
- padec (1074)
- neravnovesje (6529)
- ležanje (13465
- počasna hoja (15045)
- hitra hoja (15854)
- stanje pri miru (13421)
- sedenje (13975)

Za učenje nevronskih mrež smo uporabili 5-kratno križno validacijo. Najprej smo vse videoposnetke različnih oseb, ki jih je bilo skupaj 25, razdelili na 5 delov. Vsak del v tem koraku je tako imel 20 videoposnetkov v učni množici in 5 videoposnetkov v testni množici. Med temi 20 videoposnetki smo nato naključno izbrali še 5 videoposnetkov, ki smo jih uporabili kot validacijsko množico.

3.1 Povratna nevronska mreža tipa zaporedje v vektor

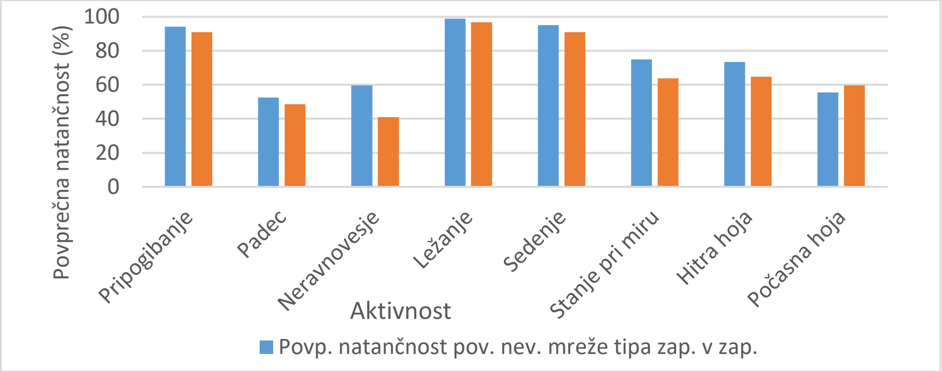
Po učenju te nevronske mreže smo jo ovrednotili na testni množici in dobili 83,2 % povprečno natančnost. Za primerjavo rezultatov smo uporabili zaporedja, ki so vsebovala izhod zadnje plasti Softmax modela Inception v3, ki smo ga učili v prejšnjem koraku. Na vsakem tako generiranem zaporedju smo nato uporabili večinsko glasovanje. Izračunali smo 77,4 % povprečno natančnost prepoznave. Graf na sliki 3 prikazuje povprečne natančnosti prepoznave posamezne aktivnosti pri uporabi zaporedij slik.



Slika 3: Povprečna natančnost prepoznavanja posamezne aktivnosti osebe pri uporabi zaporedij slik

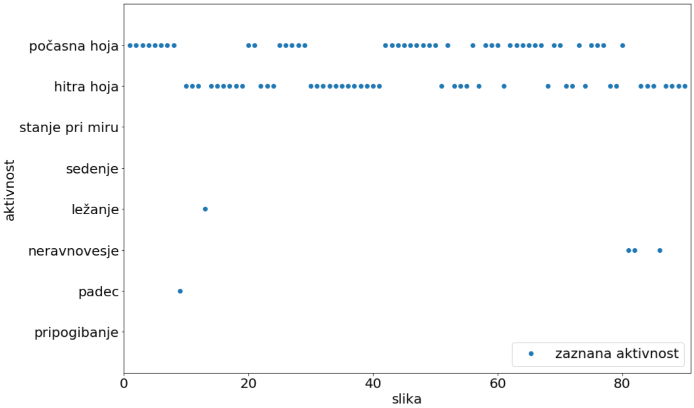
3.2 Povratna nevronska mreža tipa zaporedje v zaporedje

Povprečna natančnost prepoznavanja aktivnosti osebe s povratno nevronsko mrežo tega tipa je bila 75,5 %. Primerjavo rezultatov smo izvedli z drsnim oknom, ki smo ga premikali po zaporedjih, ki so vsebovala izhod zadnje plasti Softmax modela Inception v3, ki smo ga učili v prejšnjih korakih. Za določitev zaznane aktivnosti smo ob vsakem premiku drsnega okna uporabili večinsko glasovanje. Odločili smo se za uporabo drsnega okna velikosti 5. Dosegli smo 71,5 % povprečno natančnost prepoznavanja aktivnosti osebe. Graf na sliki 4 prikazuje povprečne natančnosti prepoznavanja posamezne aktivnosti osebe pri uporabi povratne nevronske mreže tipa zaporedje v zaporedje in drsečega okna velikosti 5.



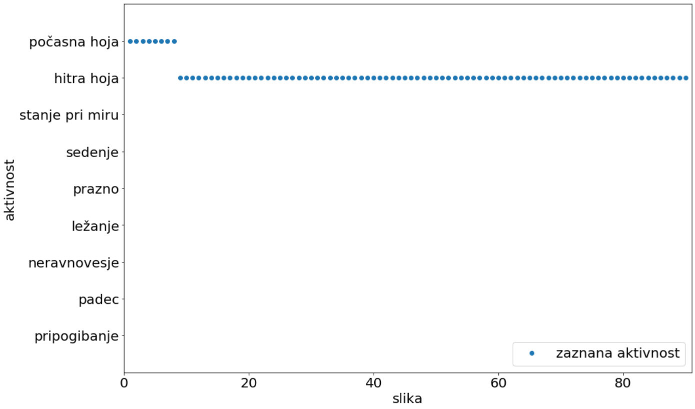
Slika 4: Povprečna natančnost prepoznave posamezne aktivnosti osebe z uporabo povratne nevronske mreže tipa zaporedje v zaporedje in drsečega okna velikosti 5

V nadaljevanju prikazujemo grafe, kjer smo primerjali delovanje različnih načinov zaznavanja aktivnosti osebe na vsaki sliki zaporedja. Na sliki 5 vidimo graf, ki prikazuje zaznano aktivnost na vsaki sliki zaporedja pri uporabi zadnje plasti Softmax modela Inception v3, ki smo ga učili prej. Na slikah 6 in 7 pa lahko vidimo rezultate, ki jih je dosegla povratna nevronska mreža tipa zaporedje v zaporedje in drsno okno velikosti 5. Vsi trije načini so zaznavali na identičnem zaporedju, ki je vsebovalo 90 zaporednih slik aktivnosti hitra hoja.



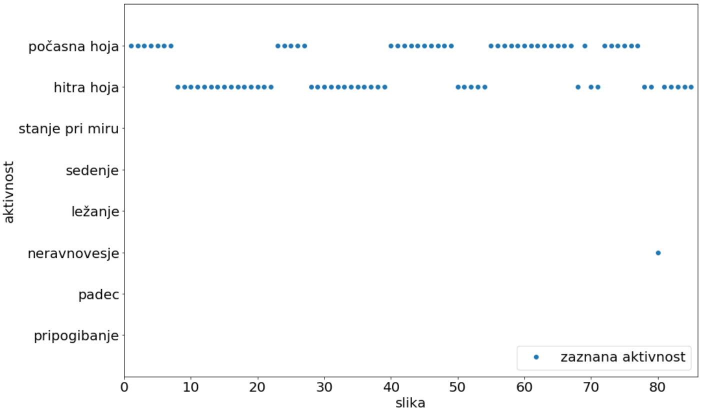
Slika 5: Primer zaznave aktivnosti hitra hoja za vsako sliko zaporedja pri uporabi izhoda zadnje plasti Softmax modela Inception v3

Najmanj uspešno je po pričakovanjih bilo zaznavanje z uporabo zadnje plasti Softmax modela Inception v3. Vidimo, da je zaznana aktivnost večinoma oscilirala med hitro in počasno hojo.



Slika 6: Primer zaznave aktivnosti hitra hoja za vsako sliko zaporedja pri uporabi povratne nevronske mreže tipa zaporedje v zaporedje

Pristop s povratno nevronsko mrežo se je izkazal za uspešnega, saj je po nekaj začetnih slikah, v nadaljevanju pravilno klasificiral vse slike kot aktivnost hitra hoja.



Slika 7: Primer zaznave aktivnosti hitra hoja za vsako sliko zaporedja pri uporabi drsnega okna velikosti 5

Drsno okno je doseglo slabše rezultate od povratne nevronske mreže. Opazimo, da je to večinoma osciliralo med aktivnostma počasna in hitra hoja.

4 Zaključek

V tem prispevku smo se ukvarjali s problemom prepoznavanja aktivnosti osebe iz zaporedja slik s pomočjo globokih nevronskih mrež. Čeprav smo eksperimentalno pokazali, da smo s pomočjo povratne nevronske mreže v povprečju natančneje prepoznavali aktivnosti osebe, menimo, da takšen pristop ni nujno najboljši za vse naloge. Zaradi manjše računske zahtevnosti so za določene naloge včasih primernejši drugi načini razvrščanja. V določenih primerih je zato smiselno kombinirati različne razvrščevalnike ali pa celo uporabiti razvrščevalnik, ki uporablja zgolj informacijo ene same slike. Zaradi opaznih razlik v natančnosti prepoznavanja posameznih aktivnostih osebe je tako pred končno odločitvijo o izbiri razvrščevalnika smiselno identificirati aktivnosti, ki jih dejansko želimo prepoznavati. Šele na tej osnovi lahko argumentirano izberemo tip razvrščevalnika.

Literatura

[1] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodriguez, E. Jauregi. Video Activity Recognition: State-of-the-Art, Sensors, 19, (2019), str. 3160.

[2] "Tensorflow detection model zoo". Dostopno na: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md. [5. 2. 2020].

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the Inception Architecture for Computer Vision. CoRR, abs/1512.00567, (2015).

[4] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, S. Schwenk, Y. Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, 2014, str. 1724–1734.

[5] Pintarič D. Prepoznavanje aktivnosti osebe iz zaporedja slik z globokimi povratnimi nevronskimi mrežami : diplomsko delo [Internet]. Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko; 2019

UPRAVLJANJE RAČUNALNIŠKE IGRE Z MISLIMI

MARKO KRAJINOVIĆ, BORUT BATAGELJ IN FRANC SOLINA
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija,
e-pošta: mk9686@student.uni-lj.si, borut.batagelj@fri.uni-lj.si, franc.solina@fri.uni-lj.si

Povzetek Z možganski vmesniki lahko ustvarjamo unikatne in inovativne načine interakcije s sodobno tehnologijo. Predstavljajo neposredno povezavo med človekom in računalnikom. V zadnjem desetletju je njihov razvoj močno napredoval. Podjetja trenutno raziskujejo njihovo uporabo na širših področjih. V namen raziskavam in izobraževanju podjetje g.tec organizira hekaton BR41N.IO. Ta omogoča razvijalcem in oblikovalcem, da se spoznajo s to tehnologijo. Jeseni sem se udeležil tega hekatona v Trbovljah in razvil projekt Imagination Solution. Prototip obsega računalniško igro v kateri uporabnik z mislimi izbira virtualne pripomočke. Z njimi si pomaga pri doseganju zastavljenega cilja.

Ključne besede:
možganski vmesnik,
Unity 3D igra,
BR4IN.IO,
hekaton
interaktivna računalniška igra.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1.7
ISBN 978-961-286-337-1

1 Uvod

Računalniški vmesniki predstavljajo točko komunikacije med človekom in računalniškim sistemom. Uporabljamo jih za opravljanje vseh interakcij z moderno tehnologijo. Ti obstajajo v različnih oblikah in vpeljejo nivo posredovanja informacij in ukazov. Možganski vmesniki omogočajo neposredno interakcijo med človeškimi možgani in računalniškimi sistemi. Z njimi lahko uporabnik upravlja sisteme zgolj s svojimi mislimi. Vmesnik meri možgansko aktivnost ter signale nato pošlje računalniškemu sistemu. Ta jih nadaljnje procesira in ovrednoti.

Trenutno smo v času velikega napredka na področju razvoja in uporabe možganskih vmesnikov. Raziskave na tem področju so začeli nevroznanstveniki že v začetkih sedemdesetih let dvajsetega stoletja [1]. Vse do začetka enaindvajsetega stoletja so ti omogočali zgolj osnovne računalniške operacije kot je premikanje miškinega kurzorja. Danes je možno z njimi opravljati veliko bolj kompleksna opravila. V zadnjem desetletju je tehnologija strojnega učenja močno napredovala. Ta predstavlja eno izmet ključnih komponent za nadaljnji razvoj programske opreme povezane z možganskimi vmesniki [2]. Zaradi takšnega napredka, uporaba te tehnologije ni omejena samo na področje medicine in nevroznanosti, ampak vključuje tudi uporabniške aplikacije, zabavno industrijo in umetnost.

2 Možganski vmesniki

Možganski vmesniki so vhodne naprave, ki povezujejo možgane z računalniškim sistemom. S specializiranimi senzorji merijo možgansko delovanje in pridobljene podatke posredujejo računalniku, ki jih običajno še nadaljnje obdela. Najpogosteje merijo elektromagnetno valovanje, ki se tvori ob aktivnosti specifičnih predelov možganov.

Vmesniki so lahko invazivni ali neinvazivni. Neinvazivni vmesniki uporabljajo senzorje v obliki EEG (elektroencefalografskih) tipal, ki se pritrdijo na določene predele človeške glave. Ti merijo elektromagnetno valovanje s površine lobanje. Ta metoda je veliko bolj primerna za večino uporabnikov. Cenovno je bolj dostopna in omogoča enostavno prenašanje vmesnika med uporabniki. Glavna slabost pri tem pristopu je bistven vpliv šuma. Signal je na površju šibkejši kot v notranjosti. Nanj pa prav tako vplivajo zunanje motnje iz okolja. Za uporabo invazivnih vmesnikov je potrebno opraviti medicinsko operacijo, pri kateri kirurg vsadi senzor neposredno v možgane. Direktna izpostavljenost senzorja omogoča bolj natančne meritve. Odziv na koristno možgansko valovanje je veliko močnejši od vpliva šuma. Z vgrajenim vmesnikom, lahko tudi stimuliramo posamezne predele možganov. Na takšen način dosežemo dvosmerno komunikacijo.

Proces delovanja možganskih vmesnikov je razdeljen na več korakov. V prvem koraku vmesnik zajame možgansko aktivnost z uporabo specializiranih senzorjev. Zagotoviti mora, da ima signal dovolj visoko časovno in prostorsko ločljivost. Prav tako je pomembno minimizirati količino šuma, da izboljšamo natančnost merjenja. Naslednji korak obsega procesiranje zajetega signala. S filtriranjem posameznih frekvenc lahko nadaljnje odstranimo morebiten zunanji šum. Prav tako pa lahko izoliramo signal le na specifične spektre možganskih valov. V tretjem koraku je zajet signal potrebno klasificirati. Z uporabo modelov strojnega učenja, klasificiramo pomen zajetih signalov glede na problemsko domeno. Za to je potrebna zadostna količina učnih podatkov. V zadnjem koraku klasificirani vhod uporabimo za opravljanje določene akcije definirane v aplikaciji.

Zaznavanje možganske aktivnosti je bolj enostavno, ko merimo odziv na znan dražljaj. Ena izmed tehnik, ki uporablja to lastnost je zaznavanje vala P300. Ta se pojavi pri človeškem odzivu na določen dražljaj. Na EEG signalu se pojavi kot skok amplitude v pozitivni smeri. Tako ga je možno preprosto zaznati in ovrednotiti. [3]

Možganski vmesniki se danes uporabljajo na številnih področjih. Pogosto se uporabljajo za raziskovanje delovanja človeških možganov. Z njimi je možno meriti odzive določenih predelov, glede na specifične mentalne aktivnosti subjekta. Prav tako imajo pomembno vlogo kot računalniški vmesnik za invalide, ki ne morejo uporabljati tradicionalnih vmesnikov. Z napredkom na področju strojnega učenja se je razširilo tudi območje uporabe možganskih vmesnikov na področjih izven raziskav in medicine. V zabavni industriji takšna tehnologija omogoča personalno prilagajanje virtualnih izkušenj. V povezavi s pametnimi napravami lahko prilagodi uporabo glede na uporabnikove preference, ne da bi ta sam aktivno skrbel za to. Takšna uporaba lahko obsega samodejno nastavljanje termostata ali svetlobe v pametni hiši, izbiro glasbe v avtomobilu ipd.

Trenutno možganski vmesniki niso še namenjeni potrošnikom. Primarno se uporabljajo v raziskovalnih ustanovah. Razvijalci se z njimi lahko srečajo v obliki razvojnih paketov, ki jih je možno naročiti od določenih proizvajalcev. Prav tako jih je pa možno zaslediti na specializiranih dogodkih, konferencah in hekatonih.

3 BR41N.IO

BR41N.IO je hekaton, ki ga organizira avstrijsko podjetje g.tec v sodelovanju z organizacijo IEEE Brain. Podjetje se ukvarja z razvojem tehnologije na področju nevroznanosti. Primarno razvijajo možganske vmesnike. Ponujajo tako invazivne kot neinvazivne metode [4]. Poleg tega ponujajo tudi izobraževanja in delavnice za razumevanje, uporabo in razvoj aplikacij z možganskimi vmesniki.

3.1 Hekaton

V okviru izobraževalne iniciative so pri podjetju g.tec razvili tekmovanje BR41N.IO. Ustanovili so ga leta 2017. Izvajajo ga večkrat letno po celem svetu [2]. Do sedaj so ga organizirali v Grčiji, Nemčiji, Sloveniji, ZDA, na Japonskem in številnih drugih državah. Dogodek je specializiran za razvoj projektov in uporabo tehnologije povezane z možganskimi vmesniki. Udeležencem zagotovijo razvojni paket, ki vključuje možganski vmesnik in vso potrebno programsko opremo za razvoj aplikacij. Dogodek je namenjen razvijalcem, oblikovalcem in umetnikom. Zaradi tega projekti vključujejo naloge iz inženirskega in umetniškega področja. Te obsegajo uporabo tehnologije možganskih vmesnikov v povezavi z upravljanjem robotov, integracijo s pametnimi napravami, aplikacijami za rehabilitacijo, računalniškimi igrami, 3D tiskanjem, izrisovanjem sanj ipd. Tekmovalci izberejo eno izmed njih ali si zamislijo svojo. Nato v štiriindvajsetih urah razvijejo projekt, ki običajno vključuje koncept in prototip aplikacije. Predstavijo ga mednarodni žiriji. Ta izbere najbolj izviren, inovativen in kakovostno izpeljan projekt.

3.2 BR41N.IO Trbovlje

Oktobra 2019 sem se udeležil BR41N.IO hekatona v Trbovljah. Dogodek se je odvijal v delavskem domu v okviru festivala Speculum Artium. Poleg hekatona je bila organizirana tudi razstava novomedijske umetnosti. Razstavljeni so bili umetniški projekti povezani s sodobno tehnologijo kot je navidezna resničnost, robotika, napredna senzorika in možganski vmesniki.

Tekmovanje sta vodila strokovnjaka s podjetja g.tec. Dogodek se je pričel ob desetih dopoldan s predavanjem o tehnologiji možganskih vmesnikov. Sledila je demonstracija njihovega razvojnega paketa Unicorn Hybrid Black. Tekmovanje se je uradno začelo ob trinajstih, ko je vsaka skupina prejela svoj razvojni paket. V naslednjih štiriindvajsetih urah so ekipe razvile projekte in ustvarile krajše predstavitve. Med tem sta strokovnjaka s podjetja g.tec udeležencem ponujala podporo pri uporabi vmesnika in njegovi implementaciji. Predstavitve je ocenjevala mednarodna žirija akademikov in strokovnjakov iz različnih podjetij visoke tehnologije. Med njimi je bila dr. Maryam Alimardanis z univerze v Tilburgu na Nizozemskem, Erika Mondria iz avstrijskega inštituta Ars Electronica ter minister za kulturo Zoran Poznič.

4 Vmesnik Unicorn Hybrid Black

Na tekmovanju smo pri razvoju projektov uporabljali možganski vmesnik Unicorn Hybrid Black (Slika 1). To je neinvazivni možganski vmesnik, ki meri možgansko aktivnost z uporabo tehnologije EEG. Naprava je pritrjena na lahko kapo. Ta vsebuje luknje, ki so namenjene apliciranju tipal na ustrezna mesta na uporabnikovi lobanji. Dve tipali je potrebno dodatno zalepiti na čeljust. Vmesnik se poveže na računalnik s standardom bluetooth in brezžično prenaša zajete signale. Naprava deluje dve uri preden jo je potrebno ponovno napolniti. Kakovost merjenja je možno izboljšati z uporabo priloženega gela, ki se ga nanese na posamezno EEG elektrodo. To izboljša prevodnost in zmanjša vpliv šuma. Naprava zajema signal s 24 biti pri 250 Hz. Signal nadvzorči 4096 krat in na takšen način zagotovi boljše razmerje med signalom in šumom [5].

Razvojni paket vsebuje tudi programsko opremo za upravljanje z napravo in uporabo zajetih podatkov. Ta omogoča prikaz zajema signala v realnem času. Orodje vsebuje tudi sistem za določanje kakovosti signala za posamezne elektrode. Ta grafično prikaže stanje elektrod glede na njihovo lokacijo. Z zeleno označi elektrode, ki ustrezno zaznavajo možgansko aktivnost. Meritve so najbolj kakovostne, ko je uporabnik zbran in se ne ozira na zunanje dejavnike. Prav tako mora biti čim bolj miren, ker s premikanjem uvaja dodaten šum.

Slika 1: Možganski vmesnik Unicorn Hybrid Black
Vir: https://www.unicorn-bi.com/product/unicorn-hybrid-black/

Drugi način uporabe vmesnika je z orodjem P300 speller. Ta omogoča pisanje besedila z mislimi. To doseže z ustvarjanjem dražljajev, ki jih nato zazna z analiziranjem uporabnikovih pričakovanih valov P300. Na zaslonu izriše tipkovnico. Uporabnik zbrano opazuje izbran gumb na zaslonu. Priporočljivo je, da v mislih šteje, ker na takšen način odpravi ostale misli, ki bi potencialno vplivale na ustrezno zaznavanje. Med tem na virtualni tipkovnici utripajo posamezne vrstice in stolpci na mestu tipk in za kratek moment prikažejo alternativno sliko. Vsakič ko utripa element, ki ga opazuje uporabnik, se sproži val P300. Sistem izmeri časovno ujemanje med utripanjem posameznih elementov in uporabnikovih odzivov. Na takšen način poišče element z najmanjšim odstopanjem. Možganski odzivi se med uporabniki razlikujejo. Zaradi tega je potrebno kalibrirati napravo pred uporabo. Ta deluje v obratni smeri kot zaznavanje. Uporabnik izbere sekvenco elementov in jih opazuje. Večje kot je število elementov pri kalibraciji, bolj ustrezno se naprava kalibrira.

Programska oprema razvojnega paketa omogoča tudi integracijo naprave s številnimi programskimi jeziki. Na GitHub repozitoriju razvojnega paketa so na voljo odprtokodni API-ji za C, C++, Python, MATLAB-ov Simulink in .NET API za C# [6]. Ti omogočajo neposredno uporabo signalov zajetih v živo. V programsko opremo je možno tudi integrirati orodje P300 speller. Razvijalec lahko na tipkovnici nastavi elemente po meri. Prav tako lahko spreminja čas zaznavanja posameznega elementa ter frekvenco utripanja.

5 Projekt Imagination Solution

Pri uporabi novih vmesnikov, se razvijalci pogosto ujamejo v past. Z novo tehnologijo skušajo opravljati interakcije, ki so bolj primerne za obstoječe vmesnike.

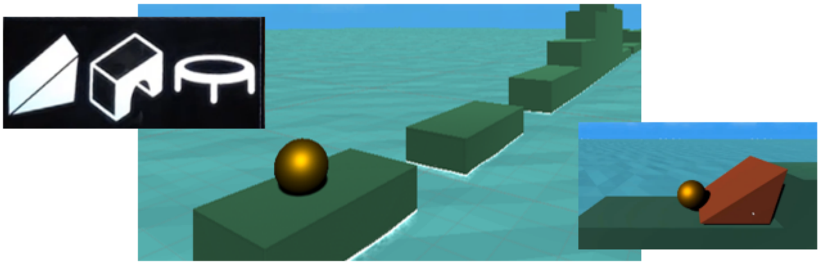
Za kakovostno rabo novih vmesnikov je potrebno uporabiti njihove unikatne značilnosti. V okviru tekmovanja v Trbovljah sem s takšnim pristopom ustvaril interaktivno 3D igro. Ta uporablja možganski vmesnik zgolj za interakcije, ki so intuitivno povezane z miselnimi procesi. Ostale interakcije uporabnik vrši s klasično uporabo tipkovnice in miške.

V okviru hekatona sem skupaj z Jonom Tavčarjem, dijakom srednje tehniške šole v Trbovljah, razvil projekt Imagination Solution. Glavna vizija projekta je interaktivna igra, kjer igralec rešuje zastavljene naloge s svojo domišljijo. Uporabnik upravlja virtualni avatar v igralnem okolju. Aplikacija mu zastavi določeno nalogo ali cilj. Uporabnikov avatar tega ne more doseči sam po sebi. Mora si pomagati z različnimi pripomočki. Te si uporabnik zamisli. Aplikacija to razbere s pomočjo možganskega vmesnika in jih ustvari v virtualno okolje. Naloge nimajo fiksnih rešitev. Vsak uporabnik jih lahko razreši na unikaten način s pripomočki, ki si jih sam zamisli. Na takšen način igra krepi kreativno mišljenje in razmišljanje izven zadanih okvirjev.

V idealni situaciji, bi aplikacija uporabila možganski vmesnik za dinamično generiranje unikatnih pripomočkov, glede na uporabnikovo zamisel. Virtualno okolje bi zgolj določilo okvirje ontologije in fizike. S kompleksnim model strojnega učenja, bi klasificirali uporabnikove zamisli na domeni pripomočkov.

Bolj enostaven pristop bi obsegal implementacijo številnih, v naprej določenih, pripomočkov. Tem bi fiksno definirali njihove funkcije. Uporabnikove zamisli bi nato klasificiral in poiskal najboljši približek iz obstoječega nabora ustvarjenih elementov. Določene parametre bi bilo možno dinamično prilagoditi in tako bolj točno predstaviti uporabnikovo idejo. To bi lahko vključevalo spreminjanje estetike posameznega elementa ali nastavljanje njegovih parametrov.

V okviru tekmovanja sem razvil prototip. Aplikacijo sem ustvaril v pogonu Unity 3D. Za povezavo z možganskim vmesnik sem uporabil integracijo orodja P300 speller. Ta pristop je omogočil implementacijo možganskega vmesnika v kratkem času, ki je bil na razpolago. Prototip vsebuje tri fiksno določene pripomočke, med katerimi izbira uporabnik. Ti vključujejo most, stopnice in odskočno ploščad. Ko jih uporabnik z mislimi izbere v orodju P300 speller, se pojavijo v virtualnem okolju (Slika 2). Nato jih uporabnik lahko premika z miško in uporabi za reševanje naloge. Zadan cilj v prototipu je pripeljati svojega avatarja do cilja 2D poligona. To je možno doseči na različne načine z uporabo implementiranih pripomočkov. Prototip predstavlja osnovno različico vizije projekta in demonstrira njegov potencial. Komisija je projektu dodelila glavno nagrado, IEEE Brain Award.



Slika 2: Vmesnik igre z možnostjo izbire pripomočkov z mislimi

Vir: svoj.

5 Zaključek

Možganski vmesniki predstavljajo inovativno povezavo med človekom in računalniškimi sistemi. V zadnjem desetletju je njihov razvoj močno napredoval z uporabo tehnik strojnega učenja. Čeprav še niso komercialno dostopni, je njihova uporaba vse bolj prisotna na področjih izven okvirjev raziskav in medicine. Trenutno smo v obdobju odkrivanja njihovega potenciala v uporabniški tehnologiji, kot so pametne naprave in zabavne aplikacije. Pri njihovi implementaciji se je pomembno osredotočiti na ustrezno uporabo, ki vključuje intuitivne miselne interakcije.

Literatura

[1] Brain Research Institute, University of California. Toward Direct Brain-Computer Communication, https://www.annualreviews.org/doi/abs/10.1146/annurev.bb.02.060173.001105.

[2] BR41N.IO Hackathon. https://www.br41n.io.

[3] Haider, A., & Fazel-Rezai, R. (2017). Application of P300 event-related potential in brain-computer interface. Event-related Potentials and Evoked Potentials. INTECH, 19-38.

[4] Podjetje g.tec. https://www.gtec.at.

[5] Unicron Hybrid Black, The brain interface, https://www.unicorn-bi.com.

[6] Unicorn Suite repozitorij. https://github.com/unicorn-bi/Unicorn-Suite-Hybrid-Black.

PREPOZNAVANJE ŠARENICE S POMOČJO NEVRONSKIH MREŽ

UROŠ POLANC IN BORUT BATAGELJ

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija,
e-pošta: mk9686@student.uni-lj.si, borut.batagelj@fri.uni-lj.si, franc.solina@fri.uni-lj.si

Povzetek Naloga obravnava pristop prepoznavanja oseb na podlagi šarenice z nevronskimi mrežami. Ideja je, da na sliki očesa pravilno detektiramo območje šarenice, s katerega nato s primernimi metodami pridobimo tako imenovan vektor značilk. Vektor značilk predstavlja kratek in unikaten opis posamezne slike. Za nevronske mreže smo uporabili klasične nevronske mreže, ki smo jim kot vhod podali vektorje značilk. Na koncu smo preizkusili še konvolucijske nevronske mreže, kjer smo kot vhod podali originalno sliko. Pri klasičnih nevronskih mrežah smo testirali večje število kombinacij metod izboljšave slike, metod izbire značilk ter nevronskih mrež. Izkazalo se je, da mreže za prepoznavanje vzorcev v kombinaciji z Gaborjevimi filtri dosegajo točnost 95,7 procenta. Pri konvolucijskih nevronskih mrežah pa se je najbolje izkazala mreža ResNet50 s točnostjo 96,4 procenta.

Ključne besede:
računalniški vid,
nevronska mreža,
segmentacija,
šarenica,
konvolucijska nevronska mreža.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1.8
ISBN 978-961-286-337-1

1 Uvod

Za razvijanje in primerjanje smo uporabili podatkovno bazo CASIA Iris Image Database Version 1.0 (CASIA-IrisV1) [1], katero sestavlja 756 slik, ki predstavljajo 108 različnih očes. Za vsako oko je bilo zajetih 7 primerov v dveh sejah. Prvo sejo sestavljajo 3 primeri, drugo pa 4. Vsaka slika je shranjena kot format BMP dimenzije 320 x 280 slikovnih elementov.

Pogledali smo si dva pristopa prepoznavanja na podlagi šarenice. Prvi obsega klasične nevronske mreže, kjer smo definirali metode segmentacije, normalizacije ter izbire značilk, katere smo nato podali kot vhod različnim nevronskim mrežam. Zadnji pristop pa uporabi konvolucijske nevronske mreže, kjer kot vhod podamo originalno neobdelano sliko iz podatkovne baze.

2 Klasične nevronske mreže

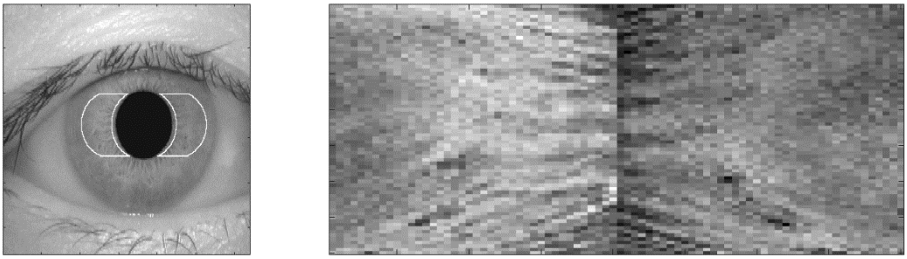
2.1 Segmentacija in normalizacija

Ker je zenica popolnoma črn predel očesa, saj se svetloba na tem delu popolnoma absorbira, lahko na sliki očesa iščemo temne predele, kar nam omogoča metoda upragovanja. Ko imamo upragovano sliko, kar še ne pomeni da smo odstranili vse druge predele, lahko nadaljujemo z binarnimi morfološkimi operacijami, ki nam pomagajo odstraniti manjše zaznane regije. S tem dobimo sliko le večjih regij, katere pa nato s pomočjo algoritmov označevanja regij ločimo in opišemo. Med ločenimi regijami izberemo površinsko največjo, saj le-ta najverjetneje predstavlja zenico. Če zenica ni največja regija, lahko sklepamo, da slika ne predstavlja očesa oziroma da je slika popačena, saj poleg zenice, obrvi ter trepalnic zelo temnih regij naj ne bi bilo.

Za metodo segmentacije smo uporabili algoritem Paula Eduarda Merlotija [2], ki predstavi nov način iskanja robov šarenice. Metoda predpostavi, da je med šarenico in beločnico dovolj razlike v odtenku barve, katero z linearnim kontrastnim filtrom še bolj poudarimo. Ker je mogoče, da so nekateri slikovni elementi znotraj šarenice zelo svetli ali zelo temni, uporabimo povprečja manjših segmentov in šele potem primerjamo spremembe v intenziteti tona. Ko je inteziteta med dvema povprečenima segmentoma dovolj velika, lahko to točko smatramo kot rob.

Metoda normalizacije katero smo uporabili je poenostavljena različica algoritma. V našem primeru dimenzijo matrike značilk takoj zmanjšamo in se osredotočimo samo na predele, ki učinkovito identificirajo posameznika. Prav tako omejimo mapiranje šarenice na območje stranic, za katera je znano, da so pod manjšim vplivom trepalnic in vek (glej Slika 1).

Pri naši poenostavitvi vzamemo prvih *n* slikovnih elementov vrstice in jih direktno shranimo v matriko stranice. To lahko storimo, saj imamo fiksno podatkovno bazo, za katero smo izračunali najmanjšo širino predela šarenice.



Slika 1: Normalizirana slika (desno) z metodami P. E. Merlotija [2]

2.2 Metode izboljšave kvalitete slike in izbire značilk

Pred izbiro značilk lahko izboljšamo kvaliteto slike z različnimi metodami, katerih cilj je sprememba podatkov, tako da le-ti izboljšajo uspešnost algorimov obdelave slike, ki sledijo. Metode lahko razdelimo v dve skupini: (1) metode prostorskih domen ter (2) metode domenskih frekvenc.

V metodah prostorskih domen (angl. spatial domain methods) se ukvarjamo neposredno s slikovnimi elementi. Vrednosti teh elementov se manipulirajo tako, da se doseže izboljšava. Pogosti sta metodi izravnave histograma (angl. histogram equalization) ter Gaborjevih filtrov (angl. Gabor filter).

Pri metodah domenskih frekvenc (angl. frequency domain methods) se slika najprej prenese v frekvenčno domeno, ponavadi s Fourierjevo transformacijo (angl. Fourier transformation), nad katero se nato izvede izboljšava, in z inverzno Fourierjevo transformacijo se dobi končna izboljšana slika. Popularni metodi domenskih frekvenc sta uporaba Gaborjeve valčne transformacije (angl. Gabor wavelet transform) in diskretne valčne transformacije (angl. discrete wavelet transform).

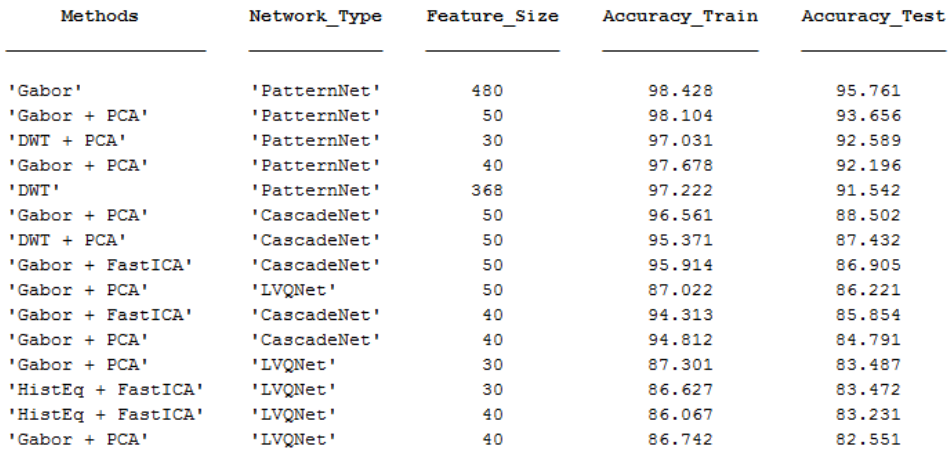
Velik problem Gaborjevih filtrov je, da imamo zelo velike dimenzije izhodnih podatkov, zato imajo tu tudi ključno vlogo metode za zmanjševanje dimenzije vektorja značilk. To lahko storimo s filtriranjem, kar je tudi razlog za ime metode Gaborjevi filtri, med katere uvrščamo vzorčenje (angl. sampling), povprečno vzorčenje (angl. average filtering), ter dekimacija (angl. downsampling). Eden izmed načinov je tudi zmanjšanje dimenzije originalne slike.

Metode kot sta Gaborjevi filtri in diskretna valovna transformacija (angl. discrete wavelet transform) imata kot izhod vektor značilk visoke dimenzije. Da zmanjšamo število dimenzij lahko uporabimo metodi kot sta analiza neodvisnih komponent (angl. independent component analysis) [3] ter linearna diskriminantna analiza (angl. linear discriminant analysis) [4].

2.3 Nevronske mreže

Za nevronske mreže smo uporabili mrežo za prepoznavanje vzorcev (angl. pattern recognition network), kaskadno usmerjeno nevronsko mrežo (angl. cascade-forward network) ter mrežo učenja vektorskih kvantizacij (angl. learning vector quantization).

Pri testiranju smo uporabili *k*-kratno križno validacijo (angl. k-fold cross validation), kjer smo poleg različnih nevronskih mrež uporabili tudi različne kombinacije izbire značilk in metod izboljšave slike, kar prikazuje Slika 2.



```
      Methods              Network_Type    Feature_Size    Accuracy_Train    Accuracy_Test
 ___________________    _____________    ____________    _______________    ______________

 'Gabor'                'PatternNet'         480             98.428              95.761
 'Gabor + PCA'          'PatternNet'          50             98.104              93.656
 'DWT + PCA'            'PatternNet'          30             97.031              92.589
 'Gabor + PCA'          'PatternNet'          40             97.678              92.196
 'DWT'                  'PatternNet'         368             97.222              91.542
 'Gabor + PCA'          'CascadeNet'          50             96.561              88.502
 'DWT + PCA'            'CascadeNet'          50             95.371              87.432
 'Gabor + FastICA'      'CascadeNet'          50             95.914              86.905
 'Gabor + PCA'          'LVQNet'              50             87.022              86.221
 'Gabor + FastICA'      'CascadeNet'          40             94.313              85.854
 'Gabor + PCA'          'CascadeNet'          40             94.812              84.791
 'Gabor + PCA'          'LVQNet'              30             87.301              83.487
 'HistEq + FastICA'     'LVQNet'              30             86.627              83.472
 'HistEq + FastICA'     'LVQNet'              40             86.067              83.231
 'Gabor + PCA'          'LVQNet'              40             86.742              82.551
```

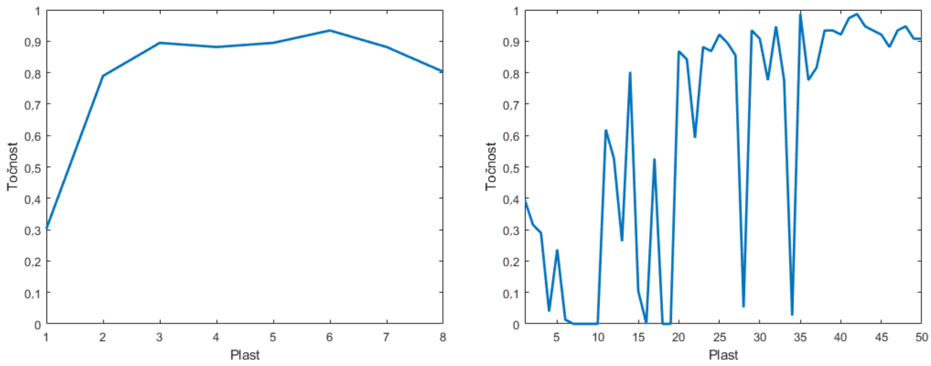
Slika 2: Najboljših pet rezultatov na posamezni nevronski mreži razvrščenih po točnosti testne množice, padajoče

3 Konvolucijske nevronske mreže

Pogledali smo tudi konvolucijski nevronski mreži AlexNet in ResNet50. Testirali smo ju na dva načina. Pri prvem smo kot vhodne podatke podali originalno sliko, pri drugem pa smo sliko segmentirali in normalizirali.

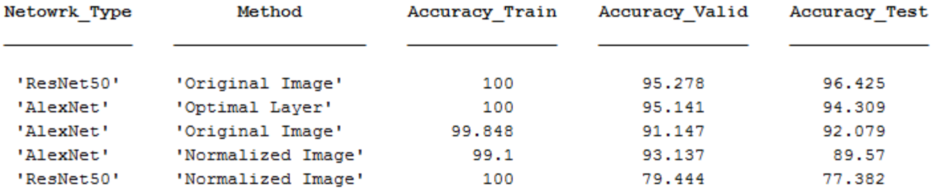
Mreži sta po sami sestavi različni in temu primerno imata različno število plasti. Zato smo se odločili tudi pogledati točnosti na vsaki plasti posebej, da bi ugotovili, ali pri večjem številu plasti lahko pride do slabših ali boljših rezultatov.

Iz Slika 3 vidimo, da z večjim številom plasti pride do večjega nihanja točnosti na posamezni plasti. To smo storili tako, da smo na vsaki plasti posebej pobrali vektor značilk za učno, testno in validacijsko množico. Z učno množico smo naučili nov model, ki uporablja metodo podpornih vektorjev (angl. support vector machine), s pomočjo katerega smo nato napovedali pričakovane razrede testne in validacijske množice. S tem smo pridobili informacijo kako posamezna plast vpliva na točnost in kako se ta točnost spremeni v primerjavi s predhodnimi plastmi. Potrebno je omeniti, da sta sliki zgrajeni s prenaučenima konvolucijskima mrežama, kjer le-ti klasificirata 1000 razredov. Podatkovna baza, ki jo uporabljamo, pa vsebuje 108 razredov, zato smo pri obeh mrežah popravili zadnjo plast, tako da le-ta pravilno klasificira našo podatkovno bazo.



Slika 3: Graf točnosti testne množice na posamezni plasti konvolucijske mreže AlexNet (levi graf) in ResNet50 (desni graf)

Kot vidimo optimalna plast ni zadnja, ampak je pri mreži AlexNet šesta, pri mreži ResNet50 pa 42. plast. To pomeni, da teoretično boljše rezultate dobimo na teh plasteh, zato smo se odločili tudi testirati točnost optimalne plasti, kjer kot vhodne podatke uporabimo originalno sliko, saj le-ta prinaša boljše rezultate. Rezultati točnosti s pomočjo konvolucijskih mrež so prikazani na sliki 4.



```
Netowrk_Type          Method            Accuracy_Train    Accuracy_Valid    Accuracy_Test
____________    ____________________    ______________    ______________    _____________

 'ResNet50'     'Original Image'             100              95.278            96.425
 'AlexNet'      'Optimal Layer'              100              95.141            94.309
 'AlexNet'      'Original Image'          99.848              91.147            92.079
 'AlexNet'      'Normalized Image'          99.1              93.137             89.57
 'ResNet50'     'Normalized Image'           100              79.444            77.382
```

Slika 4: Najboljši rezultati s pomočjo konvolucijskih nevronskih mrež

Literatura

[1] Biometrics Ideal Test, [Elektronski]. Available: http://biometrics.idealtest.org. [Poskus dostopa 4 2019].

[2] Merloti, P. E., & Swiniarski, R. (2004). Experiments on human iris recognition using error backpropagation artificial neural network. Prepared for Neural Network Class (CS533) of Spring Semester of.

[3] Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. Neural networks, 13(4-5), 411-430.

[4] Martínez, A. M., & Kak, A. C. (2001). Pca versus lda. IEEE transactions on pattern analysis and machine intelligence, 23(2), 228-233.

Latent Space Analysis of GANs for Controlled Face Deidentification

Jan Pavlin & Blaž Meden

University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, e-mail: jp8765@student.uni-lj.si, blaz.meden@fri.uni-lj.si

Abstract Each year more and more image and video data is being shared and stored on a regular basis, which brings great threat of peoples privacy. This is why face deidentification became an important topic amongst privacy and security researches. Many deidentification methods rely on pixelization or image blurring, but in recent years techniques based on formal anonymity models are replacing them. One of those possible models are generative neural networks (GNNs). In this work we analyzed the algorithm and latent space of GauGAN model. We tried to implement a method, which could be able to change attributes of generated image, based on manipulation of latent space. Results show, that it is possible to manipulate some of given attributes although we are unable to change mask of the image. We also analyzed how generated image is affected if we use k-nearest neighbours algorithm to manipulate latent space.

Keywords: gauGAN, latent space analysis, face deidentification, k-anonymity, principal component analysispintar.

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-336-4.9
ISBN 978-961-286-336-4

1 Introduction

GauGAN models is build in three parts. These are encoder, generator and discriminator.

While learning, input data is sent to encoder. Input data consists of input image (256x256x3). Encoder passes images through deep neural network and generates latent vector of length 256. Output vector is also called latent space of the image. Its values are usually distributed as a Gaussian (or Normal) distribution and normalized between 0 and 1.

Latent vector is then send to generator, where new image is generated. The generator contains a series of the SPADE (SPatially Adaptive (DE)normalization) residual blocks with upsampling layers. At every step of upsampling image, a mask of original image is also send into the generator.

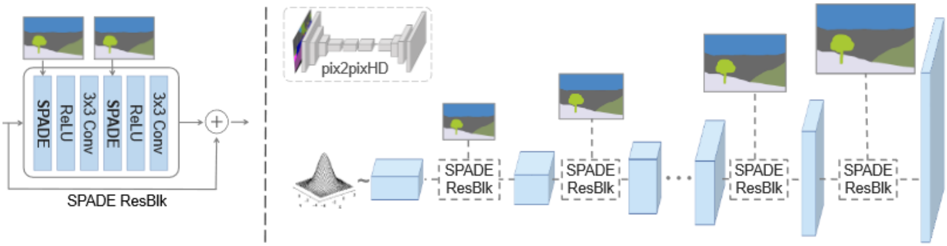


Figure 1: Generator architecture [1]

During training, a discriminator is connected to generator and it is used to predict whether generated image is similar to original. [1], [2], [3]

1.1 Related works

Similar to our project, a group of researches from University of Hong Kong [4] analyzed latent space and tried to manipulate generated images. They used different GAN model called InterFaceGAN and support vector machine (SVM) to understand how attributes impact latent space. According to the results presented in their article, they managed to change attributes in latent space and generated images that have different attributes from input.

An important difference between GauGAN and InterFaceGAN is that in GauGAN each image is generated using an additional mask, representing spatial context, based on input image. While InterFaceGAN is able to change the gender of the person in the output image with a different length/shape of hair, GauGAN can not change the mask of the image, however it is supposed to be more robust when facial shape has to be preserved. InterFaceGAN also allowed the authors to change persons pose and whether it has glasses or not, while GauGAN is able to synthesize the glasses if they are annotated as a specific category inside provided mask.

2.1 Dataset

Our GauGAN model was build on CelebAMask-HQ dataset [5]. It was build in python with library PyTorch [6]. CelebFaces Attributes Dataset (CelebA) [7] is a large-scale face attributes dataset with 30000 images, each with 40 attribute annotations. Some of most important attributes for this project were attributes of hair colour (blonde_hair, black_ hair, brown_hair), gender, lipstick and age. All of these attributes are binary. The images in this dataset cover large pose variations and background clutter.

2 Methodology and results

We have set two goals for the seminar paper. First one was the analysis of attributes and latent space, and the second was the analysis of the GauGAN algorithm using k-nearest neighbor algorithm, to demonstrate, whether it is possible to modify our generative model to utilize k-anonymity mechanisms for face synthesis.

For easier explanation of latent space and its correlation with image attributes we used visualization with PCA. PCA (Principal component analysis) is an algorithm that reduces the dimension of the latent space to smaller dimensions. We reduced the latent space (vector of length 256) into a two-dimensional vector(to be able to visualise it on 2d plane) and searched for relationships between different attributes. Using python's PyPlot library, we constructed graphs to visualize if and how certain attributes impact values of latent space. The most interesting relationship among the attributes was between the different hair colors. There are attributes for black, brown, gray, blond hair and even an attribute for baldness.

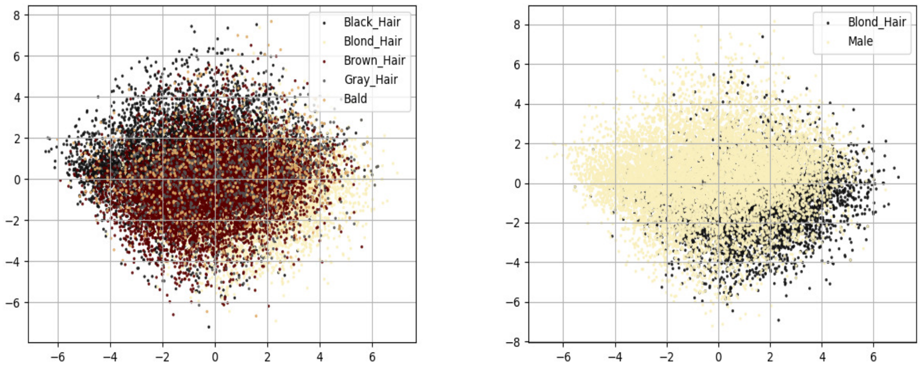


Figure 2: PCA visualization of different attributes

The Figure 2 shows different colours of hair and relation between males and blonde hair in 2d space modeled by PCA. On first graph, the biggest difference is between black and blonde hair meanwhile other hair colours are placed somewhere in between (we can notice color gradient ranging from dark to bright color tones -- covering black to blonde hair with brown tones in the middle). On second graph we can see that people with blonde hair are separated from males. Since the vizualization in Figure 2 was compressed using PCA, the attribute mapping does not represent actual attribute distribution in original space (it is only one of the possible interpretations) .

By changing the values of vectors in latent space, it is possible to change attributes of generated images. In the case of age, hair colour and lipstick, we tried to modify latent vectors to change the attributes of generated image. We used a method of modifying latent vector in multidimensional space and in 2D space generated by PCA.

In doing so, the first step was to determine how latent vectors represent each attribute. We generated latent vectors for each image in dataset and then calculated mean values of latent vectors for each and every attribute. After that, we used those mean values to calculate new value of latent vector which should represent different attributes for generated image. Some of the sample experiments were made on attributes for age and gender. To change the attribute of gender (from male to female), we have subtracted the average vector for people that were male (Male = 1) and added average vector for attribute female (Male = -1). In other words, we added difference between average male and average female vector to latent vector of the image.

$$\text{n(attr)} = \text{number of images with given attribute}$$

$$\text{average\_male} = \frac{\sum_{i=0}^{n(male)} latent\_space\_male}{n(male)}$$

$$\text{average\_female} = \frac{\sum_{i=0}^{n(female)} latent\_space\_female}{n(female)}$$

$$\text{female\_vector} = \text{average\_female} - \text{average\_male}$$

$$\text{new\_vector} = \text{original\_vector} + \text{female\_vector}$$

We also tried to change hair between original and generated images. This problem was more complex, since there were multiple attributes for hair colors. To change hair from blonde on original image to black on generated image we first subtracted the difference of average blonde and non-blonde vector (with that, the latent vector represented a non-blonde person). Then we added the average vector for black hair and generated a face. Results of the following approach on attributes of age, hair colour and lipstick are shown on Figure 3.

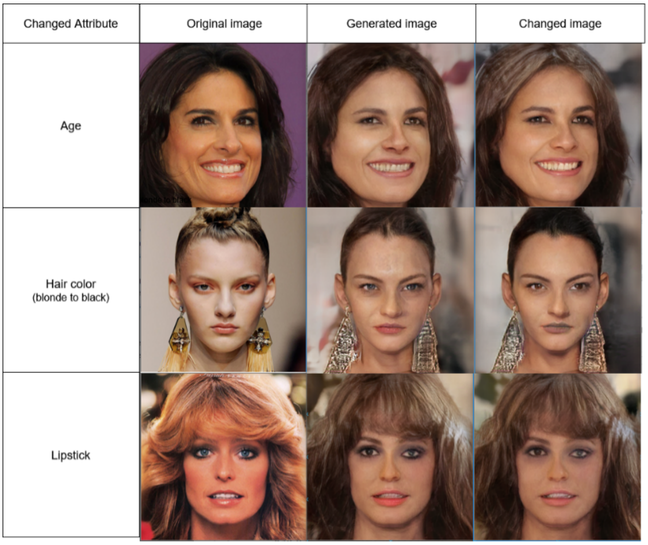


Figure 3: Changed attributes of generated images. In the last row, we subtracted lipstick attribute (since it was already present in the original image).

We can observe that some attributes of generated images have been changed. In attribute of age, shape of face generated stayed almost same but the colour of hair changed to light gray. In second row blonde hair became black and in third row we can see that lipstick was removed on last image.

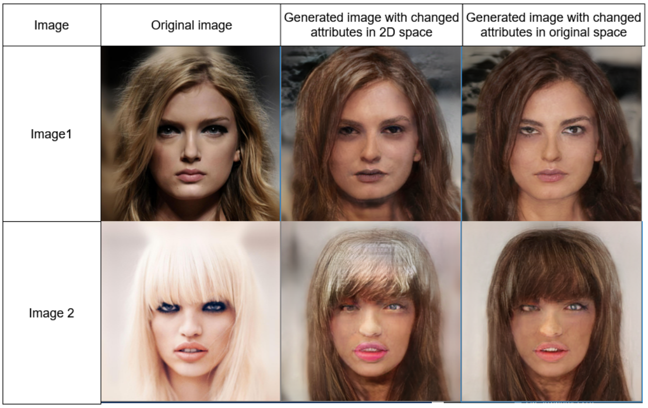


Image
Original image
Generated image with changed attributes in 2D space
Generated image with changed attributes in original space
Image1
Image 2

Figure 4: Difference between changing attributes of hair (from blonde to black) in 2d space and in original vector space of size 256.

In the Figure 4 are the results of described method to change the attributes when changing the 2D space (generated by PCA) and when changing the original multidimensional space.

It is noticeable that changing original space changes attributes better than PCA. This is most likely because the PCA generalizes multidimensional space while losing a lot of information. These results could be improved by opting for higher dimensions, but with that we would lose the ability to make simple visualizations (these are simple in 2D space).

The second objective of the seminar paper was to analyze the to analyze the possibility of incorporating k-same algorithm into our generative model. We were interested in how the model would generate faces if we used the average latent space of the k-nearest neighbors (KNN) instead of the latent space of only one image. We compared faces that were generated at k = 1, 2 and 10.

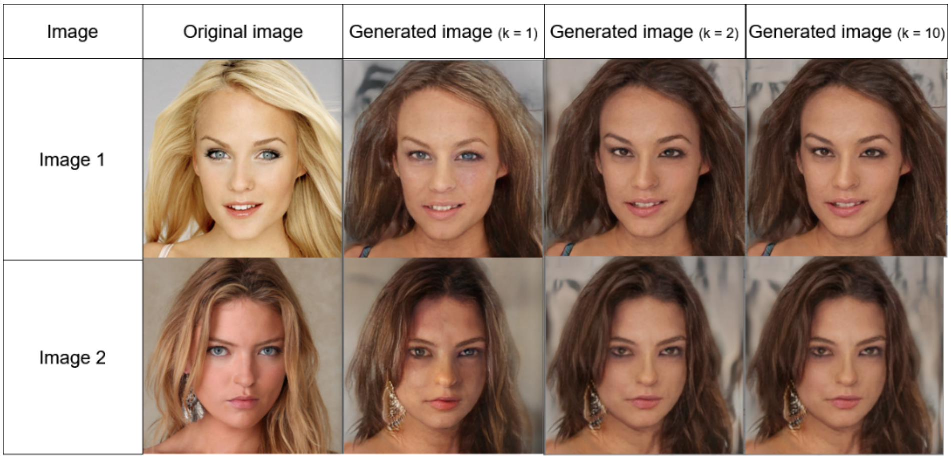


Figure 5: Differences between original, generated, generated (k=2) and generated (k=10) images

In Figure 5 we can see faces at different parameters k. Generated image (k = 1) has similar colour of hair like original image and it keep the same colour of eyes. With higher 'k' parameter faces start to slowly losing their characteristic features. At k = 10 we can see, that hair became all brown and so did eyes. This is because face becomes more generic with higher k.

3 Conclusion

In this seminar paper, we analyzed how manipulation of latent space change attributes in generated image. We have developed an approach that is based on the averaging of the embeddings (latent vectors) of the individual attribute and is able to change the appearance of the generated image.

Using original multidimensional vector performs better than PCA vector. This is mostly because PCAs two-dimensional space is generalization of original space and can lose some information during PCA data fitting.

We found that the vectors in the latent space -- although distributed by Gauss and normalized between 0 and 1 -- still retain information about the attributes of the original image. By modifying these vectors, we can force the generator to produce images with modified attributes of the original image. For example, we changed the appearance of people's ages, their lipstick, and hair color. There was a problem changing the gender, since the algorithm always gets a mask of the original image when it is generated, which also has annotated hair. Thus, when changing gender, we cannot expect hair length or shape to change.

Using the KNN algorithm, we can further mask faces (by averaging embedding values of multiple images). It is worth mentioning that generated faces will lose some information of the original image. To be more precise, they will include features of the pictures of the selected neighbors. If we set k to 10 or more, the original features of the basic image are lost by averaging.

Important problem, however, was the fact that we were using GauGAN model that was still learning. We used the model and its embeddings from 50th epoch. Using a model that has already finished its learning and the neural network has converged, we would no doubt be able to achieve even better results at changing the attributes in the latent vector or when using the KNN algorithm.

Literature

[1] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," 2019.

[2] B. Meden, Ž. Emeršič, V. Štruc, and P. Peer, "k-same-net: kanonymity with generative deep neural networks for face deidentification," *Entropy*, vol. 20, no. 1, p. 60, 2018.

[3] B. Meden, Z. Emersic, V. Struc, and P. Peer, "*κ*-same-net: Neural-network-based face deidentification," in *2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI)*. IEEE, 2017, pp. 1–7.

[4] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," 07 2019.

[5] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," *arXiv preprint arXiv:1907.11922*, 2019.

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/ 9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[7] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

EVALVACIJA KONVOLUCIJSKIH NEVRONSKIH MREŽ NA RASPBERRY PI Z USB POSPEŠEVALNIKOM GOOGLE CORAL

VID ČERMELJ, PETER PEER IN ŽIGA EMERŠIČ

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija,
e-pošta: vc5725@student.uni-lj.si, peter.peer@fri.uni-lj.si, ziga.emersic@fri.uni-lj.si

Povzetek Pri uporabi napovednih modelov, ki temeljijo na konvolucijskih nevronskih mrežah (CNN), smo pogosto omejeni na visoko zmogljive sisteme. To zmanjša vsakodnevno uporabnost, kjer bi si želeli poganjanje takih modelov na mobilnih oziroma nizko zmogljivih napravah. S tem namenom smo pregledali regijsko osnovane CNN detektorje in CNN detektorje z enim prehodom ter evalvirali naučene modele na mini računalniku Raspberry Pi 4 z uporabo USB pospeševalnika Google Coral. Najboljše rezultate smo dobili z uporabo modelov, ki so bili trenirani za izvajanje na Coral pospeševalniku z uporabo ogrodja TensorFlow Lite. Najpočasnejši povprečni čas obdelave slike ima model Faster R-CNN. Za izvajanje v realnem času sta primerni različici YOLO V2 in V3 tiny, ki omogočata obdelavo približno dveh slik na sekundo. Najboljše rezultate smo dosegli z uporabo modela MobileNet V2 SSD prilagojenega za Edge TPU. Povprečni čas obdelave slike je bil 17 milisekund, kar pomeni približno 60 sličic na sekundo.

Ključne besede:
Raspberry Pi 4
MobileNet,
R-CNN,
SSD,
YOLO,
Coral
USB
pospeševalnik.

Univerzitetna založba
Univerze v Mariboru

DOI https://doi.org/10.18690/978-961-286-337-1.10
ISBN 978-961-286-337-1

1 Uvod

Na področju video nadzora je v zadnjih letih veliko raziskav, ki za svoje rešitve uporabljajo različne mobilne naprave in mini računalnike (mobilni telefoni, RPi, NUC ipd.). Te naprave so navadno cenovno ugodne in majhne, zato so primerne za postavitev v praktično vsa okolja.

V naši analizi in evalvaciji smo se osredotočili na arhitekture, ki uporabljajo konvolucijske nevronske mreže (CNN) in imajo dobro razmerje med natančnostjo in zahtevnostjo izvajanja. Preučili smo prednosti in slabosti teh arhitektur, predvsem nam je bilo pomembno razmerje med natančnostjo napovedi ter zahtevnostjo procesiranja. Osredotočili smo se na že trenirane modele, ki smo jih pridobili s spleta. Testiranja smo izvajali na mini računalniku Raspberry Pi 4 z uporabo USB pospeševalnika Google Coral. Raziskali smo kompleksnost postavitve okolja za posamezno arhitekturo ter preučili na kakšen način je mogoče modele poganjati na Raspberry Pi s Coral pospeševalnikom oziroma s kakšnimi omejitvami se srečamo pri poganjanju modelov na napravi, ki nima namenske grafične enote.

2 Metodologija

Pregledali smo različne arhitekture konvolucijskih nevronskih mrež, ki se najpogosteje uporabljajo na področju detekcije obrazov in niso preveč računsko zahtevne. Pregledali smo vse različice posameznih arhitektur in primerjali prednosti, slabosti ter glavne izboljšave.

2.1 Detekcija objektov

Za detekcijo objektov z uporabo nevronskih mrež se za detekcijo oseb trenutno uporabljajo različne arhitekture, ki jih lahko razvrstimo v dve skupini. V prvo uvrščamo CNN detektorje z enim prehodom (angl. Single pass Convolutional Network). Primera te arhitekture sta SSD (Single Shot MultiBox Detector) [1] in YOLO (You Only Look Once) [2]. Pri testiranju smo uporabili zadnji različici te arhitekture YOLO9000 [3] in YOLO V3 [4].

V drugo skupino uvrščamo regijsko osnovane CNN detektorje (angl. Region-based Convolutional Neural Networks). V to arhitekturo spadajo R-CNN [5], Fast R-CNN [6] Faster R-CNN [7], R-FCN [8] ter arhitektura PVANET [9].

2.1.1 Regijsko osnovani CNN detektorji

Regijsko osnovane konvolucijske nevronske mreže oziroma regije s CNN značilnostmi (R-CNN) so eden izmed modernih načinov apliciranja modelov globokega učenja.

Osnovna ideja R-CNN modela je predlaganje različnih regij v sliki in izračun, če te regije vsebujejo objekt. Predloge za regije model ustvari z uporabo procesa imenovanega Selective Search [10]. S tem procesom algoritem prečeše sliko z uporabo oken različnih velikosti in poskuša združiti sosednje piksle po teksturi, barvi ali intenziteti, ki jih nato uporabi za identifikacijo objektov.

Za vsako predlagano regijo R-CNN izračuna značilke in z uporabo metode podpornih vektorjev (angl. Support-vector machine) objekte klasificira ter identificira. V zadnji stopnji R-CNN pošlje okvirne napovedi robnih škatel (angl. bounding box) skozi model linearne regresije, ki izračuna bolj natančne lokacije robnih škatel [5].

R-CNN je zaradi računske zahtevnosti neprimeren za uporabo v aplikacijah, ki delujejo v realnem času. Za taka okolja sta bolj primerna novejša Fast R-CNN in Faster R-CNN. Trenutno se izmed vseh pristopov na področju video nadzora za detekcijo objektov najbolj uporablja najnovejši Faster R-CNN [11].

V tej arhitekturi je na začetku vhodna slika podana v globoko konvolucijsko nevronsko omrežje imenovano Region Proposal Network (RPN), ki ustvari robne škatle za interesne regije (angl. Region of interest). Predlagane regije so nato preoblikovane z združevanjem ROI značilnosti iz skupnih konvolucijskih plasti. Značilke združene z ROI združevanjem Faster R-CNN uporabi za napoved ali je na nekem ROI prisoten objekt. Čas procesiranja je najbolj odvisen od števila robnih škatel, ki jih predlaga RPN [7].

Arhitektura R-FCN je podobna Faster R-CNN vendar vključuje nekaj izboljšav, ki omogočajo hitrejše izvajanje s primerljivo natančnostjo. R-FCN po ROI združevanju ne uporablja polno povezanih plasti za klasifikacijo. Vse računsko zahtevne operacije se izvedejo pred ROI združevanjem.

2.1.2 CNN detektorji z enim prehodom

YOLO je arhitektura za detekcijo objektov, ki objekte klasificira v razrede in napove kje v sliki se nahajajo prepoznani objekti.

Najprej razdeli vhodno sliko v mrežo velikosti celic S×S. Vsaka celica v mreži lahko predstavlja en napovedan objekt. Tista celica, ki vsebuje sredino objekta na koncu prevzame odgovornost za napoved objekta. Vsaka celica mreže napove največ B robnih škatel in C verjetnost razreda. Predikcija robne škatle je sestavljena iz petih vrednosti: *X* in *Y* sta koordinati središča robne škatle relativno določeni glede na celico, *W* in *H* predstavljata dimenzije škatle relativno na velikost slike. Te štiri vrednosti so normalizirane. Zadnja vrednost je ocena zaupanja napovedi (angl. confidence). Končne napovedi so kodirane v tenzor za vsako celico v mreži (Enačba 1).

$$S \times S \times (B \times 5 + C)$$

Enačba 1: Izhodni vektor napovedi za posamezno celico v mreži [2].

YOLO9000 je z uporabo normalizacije serije (angl. batch normalizaton) zmanjšal prekomerno prilagajanje podatkom (angl. overfitting). Za klasifikacijo so uporabili Darknet-19 model, ki je sestavljen iz 19 konvolucijskih plasti in petih plasti za filtriranje maksimalnih vrednosti (angl. Maxpooling layers). Največja izboljšava te različice je vpeljava sidrnih škatel (angl. anchor boxes), s katerimi mreža dobi približne podatke o položaju in dimenzijah objektov [3].

Prvi dve različici modela sta imeli probleme pri zaznavanju majhnih objektov. YOLO V3 za posamezno lokacijo v mreži naredi tri različne napovedi, kar omogoča boljše zaznavanje tako velikih kot majhnih objektov. Za predikcije razredov uporablja logistične klasifikatorje, ki omogočajo več-znakovno klasifikacijo. Glede na prejšnje različice se je povečalo tudi število konvolucijskih plasti za določanje značilk [4].

Druga arhitektura, ki jo uvrščamo med CNN detektorje z enim prehodom je SSD [1]. Osnovana je na enosmerni (angl. feed-forward) konvolucijski nevronski mreži, ki proizvede zbirko fiksne velikosti. V zbirki so podatki o robnih škatlah (angl. bounding box) in točkah, ki predstavljajo prisotnost razreda v teh škatlah. SSD združi predikcije iz različnih grafov značilk (angl. feature map) z različnimi resolucijami ter na tak način zaznava objekte različnih velikosti. SSD je preprostejši od regijsko osnovanih CNN detektorjev, saj je sestavljen samo iz enega nevronskega omrežja [1].

2.2 Ekstrakcija značilk

Za osnovno mrežo (angl. base network) smo pri arhitekturi SSD izbrali arhitekturo MobileNet. Njena prednost je hitrost računanja v primerjavi z drugimi mobilnimi arhitekturami. Osnovna mreža je v našem primeru služila za ekstrakcijo značilk, ki jih je uporabil SSD za napovedi. Glede na različico MobileNet V1 [12] in MobileNet V2 [13] je najnovejša različica MobileNet V3 [14] hitrejša in tudi bolj natančna.

Glavna prednost MobileNetV3 je uporaba t.i. AutoML tehnik (MnasNet [15] in NetAdapt [16]), ki omogočajo, da upravljavska nevronska mreža (angl. controller neural net) poišče najbolj primerno arhitekturo za dan problem. Najprej MnasNet s pomočjo tehnik okrepljenega učenja (angl. reinforcement learning) poišče grobo arhitekturo optimalne konfiguracije iz diskretnega nabora možnosti. Nato model arhitekturo natančno nastavi z uporabo NetAdapta, ki odreže malo uporabljene aktivacijske kanale v majhnih stopnjah [14].

V osrednjo arhitekturo so vključili gradnik »squeeze-and-excitation« [17], ki izboljša kvaliteto reprezentacij, ki jih izdela model. To doseže z eksplicitnim modeliranjem odvisnosti med kanali v konvolucijskih značilnostih. Avtorji so v ta namen predlagali mehanizem, ki mreži dovoljuje ponovno umerjanje značilnosti, s katerim se lahko model nauči uporabe globalnih informacij za selektivno poudarjanje informativnih značilnosti ter zavračanja manj uporabnih. Ta gradnik je v V3 postal del iskalnega prostora (angl. search space) in s tem omogočil pridobivanje bolj robustnih arhitektur.

3 Strojna oprema

Eksperimente smo poganjali na kombinaciji mini računalnika Raspberry Pi 4 in USB pospeševalnika Google Coral.

3.1 Edge Tenzor procesna enota

Edge tenzor procesna enota (Edge TPU) [18] je Googlov ASIC čip, ki je bil zgrajen za poganjanje modelov strojnega učenja z uporabo računanja na robu (angl. edge computing). Njegova prednost je majhna velikost, poleg tega porabi veliko manj energije kot TPU-ji, ki jih ima Google v svojih podatkovnih centrih.

Pri evalvaciji smo uporabili Edge TPU Coral USB pospeševalnik. Ta strojna in programska platforma omogoča gradnjo pametnih naprav s hitrim nevronskim sklepanjem (angl. neural network inferencing). Coral omogoča matematične operacije s podatki do velikosti 8-bitov. Zato da se lahko model izvaja na procesni enoti, mora biti treniran z uporabo TensorFlow kvantizacijsko zavedne tehnike (angl. quantization-aware training technique).

Coral USB pospeševalnik je velik 30 mm × 65 mm × 8 mm. V računalnik ga lahko povežemo z USB 3.0 (USB 3.1 Gen 1) kablom, ki podpira hitrosti do 5 gigabitov na sekundo. Na Coral pospeševalniku je vhod USB tipa C. Deluje lahko v dveh različnih stopnjah delovanja. Pri poganjanju na privzeti taktni frekvenci (angl. default clock frequency) mora biti temperatura zraka do 35 °C, pri uporabi maksimalne taktne frekvence (angl. maximum clock frequency) mora biti temperatura zraka do 25 °C.

Zaradi lastnosti ASIC vezij se lahko na Google Coral za detekcijo poganjajo le modeli, ki uporabljajo arhitekturo SSD in so bili kvantizirani.

3.2 Raspberry Pi 4

Za poganjanje modelov smo uporabili računalnik Raspberry Pi 4. Ima 1,5 GHz 64-biten štiri-jedrni ARMv8 procesor. 802.11ac Wireless LAN povezavnost z internetom, Bluetooth 5.0 in Bluetooth Low Energy (BLE) povezljivost. Naša različica ima 4 GB pomnilnika, 40 GPIO pinov, HDMI vhod, Ethernet vhod ter režo za Micro SD kartico.

4 Vzpostavitev okolja

Modele predstavljene v poglavju metodologija smo poiskali na spletu v formatu, ki je omogočal izvajanje na računalniku Raspberry Pi. Testirali smo z vsaj po enim modelom iz vsake arhitekture. Osredotočili smo se predvsem na modele, ki so prilagojeni za izvajanje na platformi RPi.

Največ težav pri postavitvi okolja smo imeli pri Faster R-CNN, ki se izvaja z uporabo OpenCV knjižnice. Knjižnico je bilo potrebno zgraditi direktno iz programske kode, saj jo je le na ta način možno uporabljati v C++ programih.

Za izvajanje YOLO modelov smo na sistem namestili Darknet [19]. Zaradi lastnosti naše programske opreme, računanje z osnovno različico tega ogrodja ni bilo mogoče, saj RPi nima namenske grafične enote. Z uporabo knjižnic, ki omogočajo uporabo ogrodja Darknet na napravah brez grafičnih enote nam je uspelo vzpostaviti delujoč sistem, ki smo ga testirali s programom v programskem jeziku C++.

Najmanj težav pri namestitvi smo imeli pri vzpostavitvi okolja za modele MobileNet SSD in uporabi Coral USB pospeševalnika. Vse potrebne knjižnice smo namestili z orodjem APT (Advanced Package Tool) [20], ki so nam omogočale zelo enostavne poganjanje s programskim jezikom Python.

4.1 Evalvacija

V trenutni različici TensorFlow Lite ne dopušča treniranja in poganjanja regijsko osnovanih CNN modelov in modelov YOLO. Coral USB pospeševalnik smo tako uporabili le pri evalvaciji arhitekture SSD.

Za računanje časa, ki ga model porabi za analizo vhodne slike (angl. inference time) smo uporabili kodo, ki je nameščena ob vzpostavitvi okolja za posamezne modele. Za MobileNet SSD smo inferenco merili v programskem jeziku Python, YOLO ter Faster-RCNN pa v programskem jeziku C++. Razlike zaradi programskih jezikov niso vštete v končne rezultate.

Evalvacijo arhitekture YOLO smo izvedli na različicah YOLO V2 tiny in YOLO V3 tiny. Ločeno smo izmerili povprečen čas obdelave pri prvi sliki, ko se model naloži in pri vseh nadaljnjih obdelavah.

Evalvacijo arhitekture SSD smo izvedli z različicama MobileNet V2 in MobileNet V3. Pri V2 različici smo testirali model, ki je prilagojen za Coral EdgeTPU in takega, ki vse procesiranje izvaja na RPi Za MobileNet V3 SSD na spletu še ni dostopnih modelov, ki bi bili prilagojeni za EdgeTPU, zato smo testirali model brez uporabe Coral pospeševalnika.

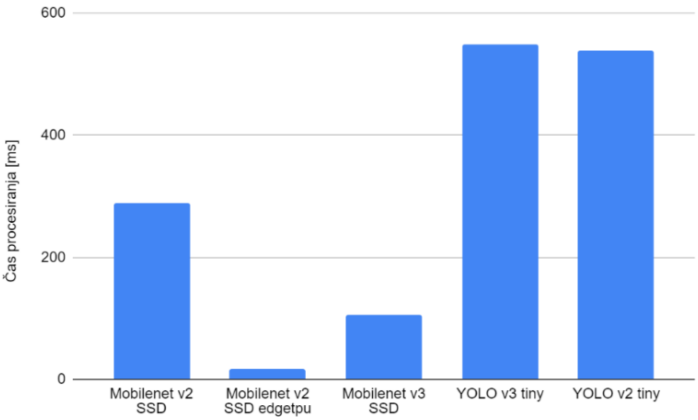
5 Rezultati

Vse modele smo testirali z dvema vhodnima slikama velikosti 1280 × 720 (HD), na slikah je bilo različno število ljudi, na eni sliki so bile štiri osebe, na drugi ena.

Uporaba Coral USB pospeševalnika je pospešila procesiranje slik za faktor 10. Na grafu 1 so prikazani povprečni časi obdelave za vse preizkušene modele. Podani so v milisekundah, pridobili smo jih s povprečjem pridobljenim iz 1000 različnih časov obdelav. Vsi modeli iz konvolucijskih nevronskih mrež z enim prehodom, ki smo jih testirali, so primerni za uporabo v realnem času. Za obdelavo videa (> 25 slik na sekundo) je primeren le MobileNet V2 SSD, ki je prilagojen za Coral pospeševalnik. MobileNet V3 SSD različica bo predvidoma še hitrejša, vendar zaenkrat še ni dostopnih modelov, ki bi bili prilagojeni za izvajanje na Coral pospeševalniku.

Bolj podrobni rezultati za testirane modele se nahajajo v tabeli 1. Inferenca ob prvi obdelavi je pri vseh modelih nekoliko počasnejša, saj se takrat v spomin naložijo uteži modela, vendar to ne vpliva na nadaljnjo obdelavo. Inferenco za posamezni model smo izračunali iz 10 različnih meritev. Povprečna natančnost uporabljenih modelov je pridobljena iz povezav, kjer smo modele pridobili in ni preizkušena z uporabo testov.

Za Faster R-CNN nam ni uspelo pognati modela v našem testnem okolju. Avtor modela, ki smo ga uporabili za evalvacijo, je dosegel povprečni čas obdelave 26 sekund za eno sliko. Modeli, ki uporabljajo R-CNN arhitekturo torej še nekaj časa ne bodo primerni za mobilne rešitve na računalnikih kot je RPi.



Graf 1: Povprečen čas procesiranja modelov.

Tabela 1: Rezultati evalvacije. Vrednost mAP predstavlja povprečno natančnost napovedi modela na bazi slik COCO.

Slika	Ime modela	Inferenca na prvi sliki	Inferenca na ostalih slikah	mAP
1 oseba	Mobilenet V2 SSD - COCO Quant Postprocess	302 ms	289 ms	22
4 osebe	Mobilenet V2 SSD - COCO Quant Postprocess	306 ms	295 ms	22
1 oseba	Mobilenet SSD V2 - COCO Quant Postprocess Edge TPU	41 ms	17 ms	22
4 osebe	Mobilenet SSD V2 - COCO Quant Postprocess Edge TPU	44 ms	18 ms	22
1 oseba	Mobilenet V3 SSD COCO	153 ms	106 ms	24.3
4 osebe	Mobilenet V3 SSD COCO	149 ms	106 ms	24.3
1 oseba	YOLO V2 tiny COCO	671 ms	539 ms	-
4 osebe	YOLO V2 tiny COCO	666 ms	537 ms	-
1 oseba	YOLO V3 tiny COCO	712 ms	548 ms	33.1
4 osebe	YOLO V3 tiny COCO	704 ms	557 ms	33.1

6 Zaključek

Raziskali smo različne CNN arhitekture, ki so se uporabljale v zadnjih petih letih. Ugotovili smo, da je za izvajanje na mini računalnikih, kot je Raspberry Pi 4, najbolje uporabiti modele, ki za detekcijo uporabljajo SSD. Podobne rezultate nam ponuja tudi arhitektura YOLO. Za obdelavo videa je potrebno uporabiti tudi Coral USB pospeševalnik, ki nam omogoča obdelavo do 60 sličic na sekundo. Ugotovili smo, da Faster R-CNN ni primeren za uporabo na RPi.

Razvijalcem, ki bi v svojih aplikacijah želeli uporabiti eno izmed obravnavanih arhitektur najbolj svetujemo uporabo Tensorflow MobileNet SSD modelov. Ta arhitektura omogoča najlažjo postavitev okolja in zelo hitro zaznavanje objektov na sliki, še posebej če sistem vključuje tudi Coral USB pospeševalnik.

Opombe

Na spodnjih povezavah se nahajajo modeli in ogrodja, ki smo jih uporabili za evalvacijo.
https://github.com/Qengineering/Faster_RCNN_ncnn
https://github.com/shizukachan/darknet-nnpack
https://github.com/digitalbrain79/darknet-nnpack
https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
https://pjreddie.com/darknet/yolo/
https://pjreddie.com/darknet/yolov2/
https://coral.ai/models/

Literatura

[1] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” *ArXiv151202325 Cs*, vol. 9905, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *ArXiv150602640 Cs*, May 2016.

[3] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *ArXiv161208242 Cs*, Dec. 2016.

[4] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *ArXiv180402767 Cs*, Apr. 2018.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *ArXiv13112524 Cs*, Oct. 2014.

[6] R. Girshick, “Fast R-CNN,” *ArXiv150408083 Cs*, Sep. 2015.

[7] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *ArXiv150601497 Cs*, Jan. 2016.

[8] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *ArXiv160506409 Cs*, Jun. 2016.

[9] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, "PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection," *ArXiv160808021 Cs*, Sep. 2016.

[10] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.

[11] L. T. Nguyen-Meidine, E. Granger, M. Kiran, and L.-A. Blais-Morin, "A Comparison of CNN-based Face and Head Detectors for Real-Time Video Surveillance Applications," *2017 Seventh Int. Conf. Image Process. Theory Tools Appl. IPTA*, pp. 1–7, Nov. 2017, doi: 10.1109/IPTA.2017.8310113.

[12] Y. Zhang, Y. Zhao, M. Liu, L. Dong, L. Kong, and L. Liu, "Vision-based mobile robot navigation through deep convolutional neural networks and end-to-end learning," in *Applications of Digital Image Processing XL*, San Diego, United States, 2017, p. 74, doi: 10.1117/12.2272648.

[13] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ArXiv170404861 Cs*, Apr. 2017.

[14] A. Howard *et al.*, "Searching for MobileNetV3," *ArXiv190502244 Cs*, Nov. 2019.

[15] M. Tan *et al.*, "MnasNet: Platform-Aware Neural Architecture Search for Mobile," *ArXiv180711626 Cs*, May 2019.

[16] T.-J. Yang *et al.*, "NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications," *ArXiv180403230 Cs*, Sep. 2018.

[17] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *ArXiv170901507 Cs*, May 2019.

[18] R. Injong, "Bringing intelligence to the edge with Cloud IoT." [Online]. Available: https://www.blog.google/products/google-cloud/bringing-intelligence-to-the-edge-with-cloud-iot/. [Accessed: 18-Dec-2019].

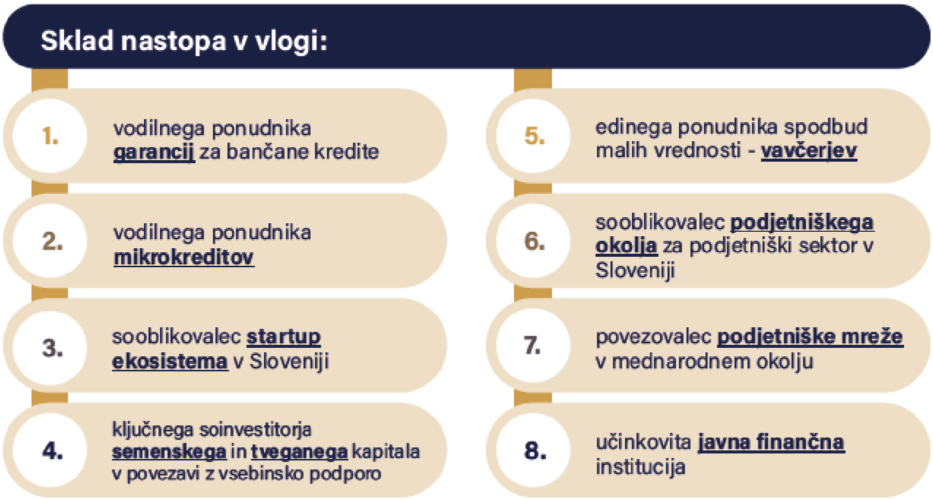
[19] "Darknet: Open Source Neural Networks in C." [Online]. Available: https://pjreddie.com/darknet/. [Accessed: 12-Jan-2020].

[20] "APT (software)," *Wikipedia*. 03-Jan-2020.

SLOVENSKI
PODJETNIŠKI
SKLAD

Javni sklad Republike Slovenije za podjetništvo
(skrajšano Slovenski podjetniški sklad)
Ulica kneza Koclja 22
2000 Maribor
T: +386 (0)2 234 12 60
E: info@podjetniskisklad.si
www.podjetniskisklad.si

Slovenski podjetniški sklad (v nadaljevanju SPS) je specializirana finančna institucija, ki mikro, malim in srednje velikim podjetjem zagonskim ter hitro rastočim podjetjem nudi učinkovite finančne in vsebinske spodbude. S svojimi aktivnostmi zapolnjuje finančne vrzeli in skupaj s finančnimi partnerji, ne le izboljša dostop, temveč tudi pogoje, ki jih morajo izpolnjevati MSP-ji pri pridobivanju finančnih virov za razvoj in rast in prodor na vedno bolj zahtevna in specializirana tržišča. SPS zaradi uspešne multiplikacije javnih virov za finančno podporo MSP-jem preko uspešnega in nadzorovanega upravljanja z javnimi viri, privablja ostale bančne in privatne vire v finančne linije za MSP-je. Hkrati s finančnimi spodbudami pa SPS prav tako sokreira slovenski start up ekosistem, je povezovalec različnih podjetniških mrež in svetovalnih institucij in s tem nadgrajuje finančno pomoč z vsebinsko pomočjo kot so različne strokovne storitve, informiranje, usposabljanja in mreženja za podjetja. S tem zastopa cilje Evropske komisije glede podpore podjetništva, rasti raziskav, razvoja in zaposlovanja.



Sklad nastopa v vlogi:

1. vodilnega ponudnika garancij za bančane kredite
2. vodilnega ponudnika mikrokreditov
3. sooblikovalec startup ekosistema v Sloveniji
4. ključnega soinvestitorja semenskega in tveganega kapitala v povezavi z vsebinsko podporo
5. edinega ponudnika spodbud malih vrednosti - vavčerjev
6. sooblikovalec podjetniškega okolja za podjetniški sektor v Sloveniji
7. povezovalec podjetniške mreže v mednarodnem okolju
8. učinkovita javna finančna institucija

Z različnimi oblikami finančnih kot vsebinskih spodbud krepimo rast in razvoj slovenskih MSPjev v vseh fazah razvoja, od zagona, do vstopa na trg, globalne rasti ter tekočega poslovanja oz. faze zrelosti. Preko SPSa so MSPji deležni do lažjega, cenejšega in hitrejšega dostopa do ugodnih finančnih virov na trgu in do bogate vsebinske podpore. Za še večjo konkurenčnost MSPjev pa SPS sodeluje tudi s Skladom skladov v sodelovanju s katerim ponuja zelo ugodna mikroposojila in semenski kapital za preboj inovativnih podjetij.

Spremljajte razpise in preverite ugodnosti posameznih produktov na:
http://www.podjetniskisklad.si/sl/razpisi.
Prijavite se na e-novice in obveščali vas bomo o novostih!

»RASTEMO SKUPAJ«

Je moto, ki velja tako za notranjo kulturo SPSa kot tudi za odnos do naših strank – MSPjev in do naših partnerjev v podjetniškem podpornem okolju.

REPUBLIKA SLOVENIJA
MINISTRSTVO ZA GOSPODARSKI
RAZVOJ IN TEHNOLOGIJO

EVROPSKA UNIJA
EVROPSKI SKLAD ZA
REGIONALNI RAZVOJ
NALOŽBA V VAŠO PRIHODNOST

EVROPSKA UNIJA
EVROPSKI STRUKTURNI
IN INVESTICIJSKI SKLADI
NALOŽBA V VAŠO PRIHODNOST

Sklad skladov

Sofinanciranje Republike Slovenije in EU iz Evropskih strukturnih in investicijskih skladov

Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

LSPO

Laboratorij za sistemsko
programsko opremo

https://rosus.feri.um.si/rosus2020
rosus.feri@um.si