RECPAD 2018

24th PORTUGUESE CONFERENCE ON PATTERN RECOGNITION

Proceedings

October 26, 2018
Coimbra, Portugal

UNIVERSITY OF COIMBRA

APRP

UNIVERSIDADE D COIMBRA

Contents

RECPAD 2018

RECPAD is the annual Portuguese Conference on Pattern Recognition, sponsored by APRP (Portuguese Association for Pattern Recognition). It is a one-day conference with an invited keynote speaker and poster sessions along the day.
This year, RECPAD2018 will be held at the Centro Cultural D. Dinis, in the heart of the UNESCO World Heritage site of the University of Coimbra , on October 26th, 2018.

Please feel extremelly welcome!

Conference Topics

RECPAD 2018 aims to promote the collaboration between the Portuguese scientific community in the fields of Pattern Recognition, Image Analysis and Processing, Soft Computing, and related areas, including, but not limited to:

- Biometrics
- Character recognition
- Classification clustering ensembles and multi-classifiers
- Data mining and big data
- Feature extraction, discretization and selection
- Fuzzy logic and fuzzy image processing
- Gesture recognition
- Hybrid methods
- Image description and registration
- Image enhancement and restoration
- Image segmentation
- Deep Learning
- Image understanding
- Information theory
- Intelligent systems
- Machine vision
- Neural network architectures
- Object recognition
- Pattern recognition applications
- Sensors and sensor fusion
- Soft computing techniques
- Statistical methods
- Syntactical methods
- Transfer Learning

Sponsors & Partners

CÂMARA MUNICIPAL DE COIMBRA

Committees

Organizing Committee

Bernardete Ribeiro, University of Coimbra

Hélder Araújo, University of Coimbra

Catarina Silva, Polytechnic Institute of Leiria

César Teixeira, University of Coimbra

Joel Arrais, University of Coimbra

Joana Costa, Polytechnic Institute of Leiria

Francisco Antunes, University of Coimbra

Program Committee

Ana Oliveira Alves, Polytechnic Institute of Coimbra

André Victor Alvarenga, Inmetro

António Dourado, University of Coimbra

António Eduardo de Barros Ruano, Universidade do Algarve

António Neves, University of Aveiro

António Pinheiro, University of Beira Interior

Armando J. Pinho, University of Aveiro

Augusto Silva, University of Aveiro

Aurélio Campilho, University of Porto

Beatriz Sousa Santos, Universidade de Aveiro/IEETA

Bernardete Ribeiro, University of Coimbra

Carlos Pereira, ISEC

Catarina Silva, Polytechnic Institute of Leiria / CISUC

César Texeira, University of Coimbra

Fernando Monteiro, Polytechnic Institute of Bragança

Filipe Rodrigues, Technical University of Denmark - DTU

Francisco Antunes, University of Coimbra

Francisco Camara Pereira, Technical University of Denmark - DTU

Helder Araújo, University of Coimbra

Hélder Oliveira, University of Porto

Hugo Proença, Univeristy of Beira Interior

Irene Pimenta Rodrigues, Universidade de Évora

Jaime Cardoso, University of Porto

Jaime Santos, University of Coimbra

Joana Costa, Polytechnic Institute of Leiria

João Rodrigues, University of the Algarve

Joaquim Pinto Da Costa, University of Porto

Joel Arrais, University of Coimbra

Jorge Barbosa, University of Porto

Jorge S. Marques, IST / ISR

Jorge Santos, ISEP

Jorge Torres, Academia Militar

José Silva, Academia Militar

Luís A. Alexandre, UBI and Instituto de Telecomunicações

Luís Teixeira, University of Porto (FEUP)

Mário Antunes, Polytechnic Inst. Leiria / INESC-TEC, CRACS, Univ. Porto

Mário Figueiredo, IT / IST

Miguel Coimbra, FCUP

Miguel Correia, Universidade de Lisboa

Noel Lopes, IPG - CISUC

Nuno Martins, ISEC - IPC

Nuno Rodrigues, IT - Instituto Politécnico de Leiria

Paulo Salgado, Universidade de Trás-os-Montes e Alto Douro

Pedro Martins, ISR - University of Coimbra

Pedro Pina, Universidade de Lisboa

Pedro Salgueiro, Dep. Informática / CENTRIA, Universidade de Évora

Petia Georgieva, University of Aveiro

Rui Gomes, University of Coimbra

Samuel Silva, DETI / IEETA - Universidade de Aveiro

Sérgio Faria, IT - Instituto Politécnico de Leiria

Thomas Gasche, Academia Militar

Verónica Vasconcelos, ISEC-IPC

Message from the General Chair

Welcome to the 24^{th} Portuguese Conference on Pattern Recognition (RECPAD2018) held at the University of Coimbra on 26th October. RECPAD is the annual Portuguese Conference on Pattern Recognition, sponsored by Portuguese Association for Pattern Recognition (APRP).
RECPAD2018 will be held at the Centro Cultural D. Dinis, in the heart of the UNESCO World Heritage site of the University of Coimbra, on October 26th, 2018. It is a one-day conference including an Invited Talk and Poster Sessions presentation.
This year, we are honoured with the presence of Prof Vincenzo Piuri a notable invited keynote speaker who will present a talk entitled "Advanced Biometric Technologies". The Conference has received about 45 papers overall distributed by sessions along the day. According to the thematic topics the papers were organised in 3 Sessions:

Session 1: Deep Learning and Health
Session 2: Genome/Drugs, Physical/HW Systems and Methods
Session 3: Computer Vision, Forecast, Social and Music Applications

Our Association APRP aims at the promotion of progress and knowledge in the area of pattern recognition, stimulating interdisciplinary interactions in various fields of science, technology, research and teaching. RECPAD is an initiative that contributes to the development and enhancement of the area of pattern recognition and its strengthening and densification at national level.
We believe that papers to be presented at the conference fulfil in part the goals of our association contributing to widen the national capabilities in the areas and to also to foster informal networking among academics and researchers.
I would like to thank everyone who collaborated with the Organization. A special word of gratitude to the members of the Technical Programme Committee for the thorough and timely review of submitted manuscripts, and also to sponsors for the invaluable support. Recognition and a deep thank must also go to the members of the Organizing Committee who worked hard for the success of this conference.
I welcome all participants and I hope you enjoy RECPAD2018 and your stay in Coimbra.

Bernardete Ribeiro,
General Chair

Invited Speaker

Vincenzo Piuri

Università degli Studi di Milano
Friday, 26th October, 2018
FRI2 – Plenary Session
10:15-11:15
(Centro Cultural D. Dinis)
Chair: Bernardete Ribeiro

Bio: Vincenzo Piuri has received his Ph.D. in computer engineering at Politecnico di Milano, Italy (1989). He is Full Professor in computer engineering at the Università degli Studi di Milano, Italy (since 2000). He has been Associate Professor at Politecnico di Milano, Italy and Visiting Professor at the University of Texas at Austin and at George Mason University, USA.
His main research interests are: intelligent systems, signal and image processing, machine learning, pattern analysis and recognition, theory and industrial applications of neural networks, biometrics, intelligent measurement systems, industrial applications, fault tolerance, digital processing architectures, and cloud computing infrastructures. Original results have been published in more than 400 papers in international journals, proceedings of international conferences, books, and book chapters.
He is Fellow of the IEEE, Distinguished Scientist of ACM, and Senior Member of INNS. He has been IEEE Vice President for Technical Activities (2015), IEEE Director, President of the IEEE Computational Intelligence Society, Vice President for Education of the IEEE Biometrics Council, Vice President for Publications of the IEEE Instrumentation and Measurement Society and the IEEE Systems Council, and Vice President for Membership of the IEEE Computational Intelligence Society.
He is Editor-in-Chief of the IEEE Systems Journal (2013-19), and Associate Editor of the IEEE Transactions on Computers and the IEEE Transactions on Cloud Computing, and has been Associate Editor of the IEEE Transactions on Neural Networks and the IEEE Transactions on Instrumentation and Measurement.
He received the IEEE Instrumentation and Measurement Society Technical Award (2002). He is Honorary Professor at Obuda University, Budapest, Hungary, Guangdong University of Petrochemical Technology, China, Muroran Institute of Technology, Japan, and the Amity University, India.

“Advanced Biometric Technologies”

Biometrics concerns the study of automated methods for identifying an individual or recognizing an individual among many people by measuring one or more physical or behavioral features. Certain physical human features or behaviors are characteristics that are specific and can be uniquely associated to one person. Retinas, iris, DNA, fingerprint, palm print, or pattern of finger lengths are typical physical features that are specific to individuals. Also the voice print, gait, or handwriting can be used to this purpose.
Nowadays biometrics is rapidly evolving. This science is getting more and more accurate in recognizing and identifying persons and behaviors. Consequently, these technologies become more and more attractive and effective in critical applications, such as to create safe personal IDs, to control the access to personal information or physical areas, to recognize terrorists or criminals, to study the movements of people, to monitor the human behavior, and to create adaptive environments. The use of biometrics in the real life often requires very complex signal and image processing and scene analysis, for example encompassing biometric feature extraction and identification, individual tracking, face tracking, eye tracking, liveness/anti-spoofing tests, and facial expression recognition. This talk will review the main biometric traits and analyze the opportunities offered by biometric technologies and applications to support a broad variety of applications. Attention will be given to the current trends in research and applications.

Poster Session 1

Deep Learning and Health

Paper:#4

Generalization Performance of Convolutional Neural Networks for Heart Sound Segmentation

By: Renna, F. Oliveira, J. Coimbra, M.

Generalization Performance of Convolutional Neural Networks for Heart Sound Segmentation

Francesco Renna
frarenna@dcc.fc.up.pt
Jorge Oliveira
oliveira_jorge@dcc.fc.up.pt
Miguel T. Coimbra
mcoimbra@dcc.fc.up.pt

Instituto de Telecomunicações
Faculdade de Ciências da Universidade do Porto
Porto, Portugal

Abstract

In this paper, deep convolutional neural networks are used to segment heart sounds into their main components. A further post-processing step is applied to the output of the proposed neural network, which induces the output state sequence to be consistent with the natural sequence of states within a heart sound signal (S1, systole, S2, diastole).

The generalization performance of the proposed approach is assessed by training the algorithm on the PhysioNet dataset and testing it on the DigiScope dataset. The proposed solution achieves an average sensitivity of 73% and an average positive predictive value of 82.5% in detecting S1 and S2 sounds.

1 Introduction

Cardiac auscultation is arguably the most cost-effective first line of screening for a large number of heart conditions. On the other hand, heart sounds are difficult to identify and analyze by the human listener, as their frequency content is at the lower end of the audible frequency range. These reasons have motivated recent research efforts in automatizing the analysis of phonocardiogram (PCG) signals, in order to extract useful diagnostic information from them.

A key step required in the analysis of PCG signals is represented by the segmentation of heart sounds in their fundamental components. In fact, each heart cycle is normally divided into a first heart sound (S1), a systolic interval, a second heart sound (S2), and a diastolic interval (See Fig. 1).

Several solutions have been proposed in the literature to perform PCG segmentation (see [4] for a general overview). A first class of segmentation algorithms is based on the use of peak-picking algorithms to estimate the principal heart sounds S1 and S2, as well as their boundaries [3]. A second class of segmentation algorithms leverages statistical models as the hidden semi-Markov model (HSMM) to include prior information about the sequential nature of PCG signals. Recently, HSMMs have been shown to provide state-of-the-art results by introducing explicit modeling of the statistics of the time spent by the PCG signal in each state [7]. A third class of segmentation algorithms is based on the extraction of features from the PCG, which are then assigned to the different heart sound states using a classifier. Some of the classifiers used for heart sound segmentation include support vector machines (SVMs) [8] and, more recently, deep neural networks [1].

This work studies a novel heart sound segmentation approach, which is based on the use of a deep convolutional neural network (CNN). The proposed method, unlike the existing deep learning solutions, does not relies on the extraction of *ad hoc* features from the signal. On the other hand, it can be applied directly to the PCG signal itself or to envelograms extracted from it. In this way, the sounds features that minimize segmentation errors are learnt directly from training data by the CNN.

The proposed segmentation approach involves the following steps: i) pre-processing of the PCG signal and extraction of envelograms from it; ii) application of a trained CNN to different portions of the envelograms extracted from the PCG signal; iii) combination of the CNN outputs corresponding to the different portions of the PCG, in order to produce the estimated state sequence.

2 Methods

In this section, the three main steps of the proposed segmentation algorithm are described in details. The proposed approach consists in a training phase and a testing phase. Signals involved in both training and testing are pre-processed according to the methods described in Section 2.1. Labeled training data are used to determine the parameters (weights) that define the operations implemented by the CNN. In the testing phase, the trained CNN is applied to the pre-processed testing data, and the corresponding output undergoes a further post-processing stage in order to generate the estimated state sequence associated to each heart sound in the testing set.

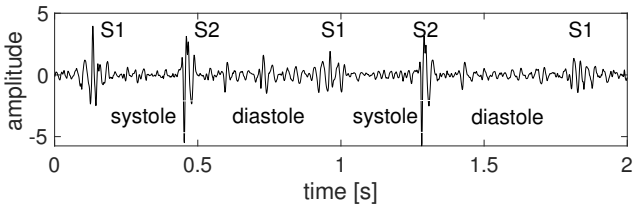


Figure 1: Example of a recorded PCG signal in which the following four main heart sound components are shown: S1, systole, S2, and diastole.

2.1 Pre-processing

PCG signals are first filtered with high-pass and low-pass Butterworth filters of order two with cut-off frequencies equal to 25 Hz and 400 Hz, respectively. The four envelograms/envelopes considered in [7] are extracted from the filtered signals: i) homomorphic envelogram, ii) Hilbert envelope, iii) wavelet envelope, and iv) power spectral density (PSD) envelope. Such envelograms are then downsampled at 50 Hz [7]. Finally, all the four envelograms/envelopes are normalized in order to have zero mean and unit variance.

For each heart sound, the normalized envelograms are collected in the 4-dimensional signal $\mathbf{x}(t)$, where $\mathbf{x}(t) \in \mathbb{R}^4$ for $t = 0, \ldots, T-1$, and where t indicates the time instant. Then, $s(t)$ is defined as the sequence containing the state labels associated to each time instant, i.e., $s(t) \in \{0,1,2,3\}$, where state 0 corresponds to an S1 sound, state 1 to a systole interval, state 2 to an S2 sound, and state 3 to a diastole interval. Then, given a heart sound signal $\mathbf{x}(t)$, the objective of the segmentation algorithm is to provide an estimate of the corresponding state sequence $s(t)$.

Four-dimensional patches of fixed length n are extracted from the signal $\mathbf{x}(t)$, with a given stride τ. Such portions of the signal $\mathbf{x}(t)$ represent the inputs of the CNN.

2.2 Convolutional neural network architecture

Various CNN architectures have been presented in the quickly growing deep learning literature. This work proposes the use of a CNN architecture which is inspired by the U-net originally presented in [6] for image segmentation. The proposed architecture is reported in Fig. 2.

Such deep network contains convolutional layers which are followed by rectified linear unit (ReLU) activation functions. Each filter in a convolutional layer is defined by 3 weights. Moreover, the proposed architecture includes max pooling layers, which are responsible for downsampling (by a factor 2) the outputs of middle layers, as well as upsampling layers (by a factor 2) which are interleaved with the late convolutional layers. Skip connections are also inserted in the network, in order to allow direct transfer of information from the first layers to the late layers. Finally, the last convolutional layer is followed by a softmax activation function, so that the CNN outputs contain the probability that each sample of the PCG input signal belongs to state 0, 1, 2, or 3. Note that the considered CNN implements an encoder-decoder architecture, in the sense that the outputs of the middle layers offer a compact representation of the input

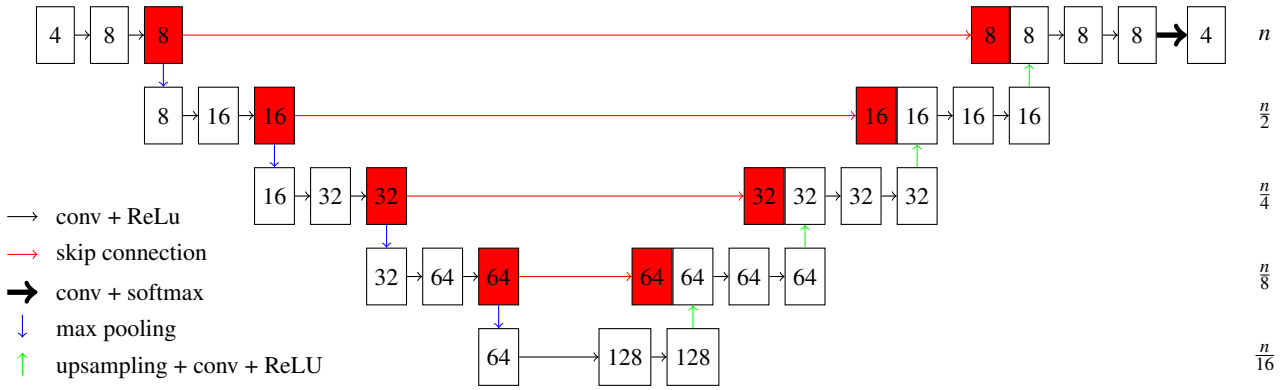


Figure 2: CNN used in the proposed segmentation algorithm. The numbers inside the boxes in the diagram represent the number of filters in the corresponding convolutional layer. The numbers on the right hand side of the figure indicate the spatial dimension of the inputs and outputs of the layers contained in the corresponding row.

signals in a low-dimensional manifold which contains the main information about the segmentation state of the PCG, thus reducing the impact of noise and signal variability.

2.3 Post-processing

The information obtained from different overlapping patches is combined by simply averaging the state probabilities associated to the CNN outputs. Then, a first estimate of the state sequence $s(t)$, which is denoted by $\tilde{s}(t)$, is obtained by choosing the state corresponding to the maximum probability among S1, systole, S2, or diastole. Then, in order to force the output sequence to contain only admissible transitions among states, a further post-processing step is performed on $\tilde{s}(t)$, which leads to the definition of the output sequence $\hat{s}(t)$ as follows: $\hat{s}(0) = \tilde{s}(0)$, and for $t > 0$,

$$\hat{s}(t) = \begin{cases} \tilde{s}(t) & , \text{ if } \tilde{s}(t) = (\hat{s}(t-1)+1) \mod 4 \\ \hat{s}(t-1) & , \text{ otherwise} \end{cases} . \quad (1)$$

3 Experiments

The proposed segmentation algorithm is compared with the method described in [7], which is currently considered as the state-of-the-art PCG segmentation algorithm.

The generalization performance of the proposed CNN-based segmentation algorithm and of the HSMM-based method in [7] is tested by training both algorithms on the publicly available PhysioNet dataset and testing them over heart sounds contained in the DigiScope dataset. The PhysioNet dataset contains 406 pathological heart sounds and 386 sounds collected from healthy patients. The DigiScope dataset contains sounds from 29 different healthy individuals, ranging in age from six months to 17 years old.

The CNN used in the proposed segmentation method is trained with patches of dimension $n = 64$ that are extracted from the training recordings with a stride of $\tau = 8$ samples. The weights of the CNN are learnt using the categorical cross-entropy loss function and the Adam optimizer with learning rate equal to 10^{-4} [2]. The maximum number of training epochs is fixed to 15, and early stopping is adopted by extracting 10% of the training data and using them for cross-validation, thus retaining the weights corresponding to the minimum loss function on cross-validation data.

Two metrics are used to evaluate the performance of the proposed algorithm in determining the position of the fundamental heart sounds S1 and S2: positive predictive value (P_+) and sensitivity (S). A true positive (T_p) is counted when the center of an S1 (S2) sound in the estimated sequence $\hat{s}(t)$ is closer than 60 ms from the center of the corresponding S1 (S2) sound in the ground truth sequence $s(t)$. All other S1 and S2 sounds in the estimated state sequence are considered as false positives (F_p).

The results obtained with the considered experimental framework are the following: the proposed segmentation method achieves an average sensitivity of 73% and an average positive predictive value of 82.5% when tested over the DigiScope dataset, whereas the method in [7] reaches an average sensitivity of 77.8% and an average positive predictive value of 80.7%. Note that, when trained and tested via 10-fold cross-validation over the PhysioNet datset, the proposed method outperforms the segmentation algorithm described in [7] both with respect to sensitivity and positive predictive value [5]. On the other hand, the generalization results obtained over the DigiScope dataset point out that further precautions need to be employed when using CNNs for heart sound segmentation in order to avoid overfitting. A possible solution could be represented by the use of data augmentation procedures to diversify further the training set.

4 Conclusions

In this work, the use of deep convolutional neural networks for heart sound segmentation was described. In particular, the generalization performance of the proposed method has been tested by training it on the PhysioNet dataset while testing it over the DigiScope dataset. Experimental results showed that the considered CNN-based segmentation algorithm achieves a larger positive predictive value than the state-of-the-art segmentation method in this experimental framework, while providing slightly lower sensitivity.

Acknowledgements

This article is a result of the project NanoSTIMA, NORTE-01-0145-FEDER-000016, supported by Norte Portugal Regional Operational Programme (NORTE 2020), through Portugal 2020 and the European Regional Development Fund (ERDF) and project UID/EEA/50008/2013. This work was also funded by the FCT grant SFRH/BPD/118714/2016.

References

[1] T. Chen et al. S1 and S2 heart sound recognition using deep neural networks. *IEEE Trans. Biomed. Eng.*, 64(2):372–380, 2017.

[2] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press Cambridge, 2016.

[3] H. Liang, S. Lukkarinen, and I. Hartimo. Heart sound segmentation algorithm based on heart sound envelogram. In *Computers in Cardiology*, pages 105–108, 1997.

[4] C. Liu et al. An open access database for the evaluation of heart sound algorithms. *Physiological Measurement*, 37(12):2181–2213, 2016.

[5] F. Renna, J. Oliveira, and M. T. Coimbra. Convolutional neural networks for heart sound segmentation. In *European Signal Processing Conference (EUSIPCO)*, pages 762–766, Sept 2018.

[6] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[7] D. Springer, L. Tarassenko, and G. Clifford. Logistic regression-HSMM-based heart sound segmentation. *IEEE Trans. Biomed.l Eng.*, 63(4):822–832, 2016.

[8] J. Vepa. Classification of heart murmurs using cepstral features and support vector machines. In *IEEE EMBC*, pages 2539–2542, 2009.

Paper:#5

Transfer Learning of ImageNet Neural Network for Pigmented Skin Lesion Detection

By: Pereira, P. Fonseca-Pinto, R. Paiva, R. Tavora, L. Assuncao, P. Faria, S.

Transfer Learning of ImageNet Neural Network for Pigmented Skin Lesion Detection

Pedro M. M. Pereira[13]
pedrommpereira@co.it.pt
Rui Fonseca-Pinto[12]
rui.pinto@ipleiria.pt
Rui Pedro Paiva[3]
ruipedro@dei.uc.pt
Luis M. N. Tavora[2]
luis.tavora@ipleiria.pt
Pedro A. A. Assuncao[12]
amado@co.it.pt
Sergio M. M. de Faria[12]
sergio.faria@co.it.pt

[1] Instituto de Telecomunicações
Portugal
[2] ESTG
Polytechnic Institute of Leiria
Portugal
[3] DEI - FCTUC
University of Coimbra
Portugal

Abstract

Traditional Artificial Neural Networks (ANN) have been investigated in the past for skin lesion classification and nowadays their performance is already quite useful to assist in medical diagnosis and decision processes. In the field of visual object recognition, recent developments of such networks (Deep Convolutional Neural Networks) are currently the winners of the ImageNet competition. This work extends the use of CNN for classification of pigmented skin lesions, by investigating a training methodology based on transfer learning on pre-trained networks.

1 Introduction

The importance of skin lesion classification arises from the fact that one of the most dangerous skin cancers, the melanoma, is developed from pigmented melanocytes [5] and its incidence in the world population is increasing very fast. Skin cancer can be either benign and malignant. Since the melanoma is malignant, it is very likely to cause death after some time. However, if diagnosed at early stages, high cure rates are achievable. Thus, early detection and full characterisation of suspicious skin lesions is the key to reduce mortality rates associated to this type of skin cancer. The development of computer vision techniques to automatically identify melanoma has been under study for decades [3] and automatic techniques for detection and classification is becoming increasingly useful to assist dermatologists and to support expert systems [6, 9].

Recent advances in visual recognition led to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [2, 10], which uses a dataset comprising more than 14 million images (of which 1 million have bounding box annotations with around 100 hundred words) that can be divided into 1000 different labels – manually validated by crowd-sourcing. The ImageNet Challenge is currently considered to be one of the most important initiatives and the dataset has therefore become a benchmark standard for large-scale object recognition, i.e., image classification, single-object location and object detection. Due to its competition-based approach, many authors are constantly improving their image classification/recognition algorithms every year. This has led to an exponential growth of related research and significant advances in state-of-the-art techniques [10].

This work focuses on studying the performance of skin cancer detection using highly-accurate networks, developed in recent years for ImageNet. Relevant comparisons are made with the performance obtained for the 1000 categories in ImageNet. To this end, the ISIC dataset [1] is selected, as the collection of skin lesion images. This dataset contains a total of 3438 images that can be divided into: 2380 benign and 1058 malignant lesions. These malignant lesions are classified as melanoma, basal cell carcinomas and squamous cell carcinoma, while the remaining ones are benign. Such classification was obtained from an unspecified number of skin cancer experts.

The transfer learning approach used in this research study uses some selected pre-trained networks from ImageNet to first extract a number of abstract features, which are fed forward to several different classifiers. Then the classification performance is evaluated and discussed.

2 Proposed Approach

The proposed approach follows a processing pipeline from the input image data to the output classification results. Firstly, before entering in the network, a pre-processing stage is responsible for performing data augmentation and then image resizing to match the network intake. Secondly, these data enters in the pre-trained network whose output is fed to the final classifier. Several classifiers are studied in this work. Different alternatives are separately trained resorting to both original data and augmented data with 20% random information holdout for later evaluation of the trained network.

2.1 Architectures

In ILSVRC history there are several pre-trained networks, already capable of image classification over 1000 different categories. This work elects 5 of the must frequently used networks, which have shown to be able to adapt to other identification and classification problems. These networks are: Alexnet [7], pioneering networking comprising 25 layers and it was the winner of the 2012 ILSVRC; VGG16 and VGG19 Net [11], reinforced the notion that convolutional neural networks must have layers in depth such that visual data present a hierarchical representation; GoogLeNet [12], has the Inception module that deviates from the standard sequential layer-stacking approach and it was the winner in 2014; and ResNet50 [4], presents an innovative way of solving the vanishing gradient problem, it comprises 177 layers and it was the winner in 2015.

2.2 Pre-processing: data augmentation and image resizing

To increase accuracy, data augmentation is performed by using a limit set of random transformations [8]. In this work the following transformations were selected: *Intensity Values Adjustment*: increases the contrast of the image; *Contrast-Limited Adaptive Histogram Equalization*: enhances the contrast of a given grayscale image by transforming the values so that its distribution matches a uniform/flat histogram (256 bins); *Random Brightness*: induces brightness variation to the image; *Random Edge-Aware Local Contrast*: enhances or flattens the image local contrasts; *Random Sharpness*: sharpens the image using the unsharp-masking method; *PCA Colour Jitter*: modifies the intensities of the RGB channels in the image according to the PCA transformation; *Random Affine Transformations*: operation between affine spaces that preserves points, straight lines and planes. As a final note, the augmentation strategies are not all used at the same time. The *PCA Colour Jitter* and *Random Affine Transformations* are always used at the end of the augmentation step, but the remaining operators are only randomly applied with a 10% change (each). After this stage, each image is augmented 200 times, thus effectively making the dataset 200 times larger.

After a possible augmentation step, and before entering the network, all input data (images) is resized to fit the network intake. Apart from Alex-Net, which receives a 277x277 (pixel) RGB image, all other networks accept a 224x244 (pixel) RGB image. Therefore, as a final step before entering the network, the images are resized to their smallest dimension (maintaining aspect ratio) and then centre-cropped to remove the outer border in excess (if any).

	AlexNet			VGG16			VGG19			GoogleNet			ResNet50		
	Acc	SE	SP	Acc	SE	SP	Acc	SE	SP	Acc	SE	SP	Acc	SE	SP
SVM	30.9	99.5	0.4	34.6	97.6	6.7	46.7	69.2	36.8	30.7	100	0.0	30.7	100	0.0
KNN	74.5	61.1	80.5	67.7	56.9	72.5	71.3	60.2	76.3	73.8	57.3	81.1	72.8	60.2	78.4
Tree	68.7	46.9	78.4	64.2	40.8	74.6	64.0	40.3	74.6	67.7	51.2	75.0	61.3	49.3	66.6
Linear	70.3	4.3	99.6	69.4	55.9	75.4	55.5	86.3	41.8	73.9	52.1	83.6	75.8	47.9	88.2
NaiveBayes	64.9	73.5	61.1	62.6	66.4	60.9	64.8	64.5	64.9	64.9	73.9	60.9	72.5	55.9	79.8

Table 1: Test Results without using Augmented Data in training

	AlexNet			VGG16			VGG19			GoogleNet			ResNet50		
	Acc	SE	SP	Acc	SE	SP	Acc	SE	SP	Acc	SE	SP	Acc	SE	SP
SVM	72.6	54.5	80.7	71.9	58.8	77.7	71.9	58.8	77.7	71.6	60.7	76.5	72.8	53.1	81.5
KNN	75.1	62.6	80.7	70.3	52.1	78.4	70.3	52.1	78.4	71.8	55.5	79.0	75.4	64.9	80.0
Tree	65.5	48.8	72.9	68.0	46.9	77.3	68.0	46.9	77.3	67.5	46.0	77.1	69.3	53.1	76.5
Linear	78.0	52.1	89.5	60.8	80.1	52.3	60.8	80.1	52.3	69.4	69.2	69.5	78.5	43.1	94.1
NaiveBayes	67.1	66.8	67.2	69.3	0.0	100	69.3	0.0	100	62.6	70.6	59.0	67.7	72.5	65.5

Table 2: Test Results using Augmented Data in the training

2.3 Learning Strategy

As mentioned before, the overall architecture includes ImageNet networks and a transfer learning scheme for feature extraction using alternative classifiers. Since the selected pre-trained architectures already provide highly accurate predictions in the ImageNet challenge, it is assumed that they are also able to extract a great variety of abstract knowledge/features from the given images containing skin lesions. In this transfer learning strategy, the output of the last convolutional layer in the pre-trained ImageNet network is connected to several alternative classifiers. The classifiers used in this work are: the SVM classifier, the K-Nearest Neighbours, the Tree classifier, a Linear classifier and a NaiveBayes classifier.

3 Results and Discussion

Using the ImageNet networks as feature extractor on the original 3438 images, while holding out 20% of this data for later testing, the network knowledge provides an average accuracy of 61% on the testing data, while the accuracy obtained in training data is 87% on average. The overall results are shown in Table 1, where it can be observed that the best performing classifier is the KNN with an average accuracy of 72% on unseen test data across the different networks and 100% on the training data. Still regarding the training data performance, the SVM and the Tree classifiers achieve accuracies of 99% and 98%, respectively. However, only 62% and 61% accuracy is obtained on unseen test data.

When data augmentation is used, the performances increase by 9% on the test-set and lose 12% accuracy on the training-set. Table 2 is presented for comparison with the previous results. In this case the training-set only comprises augmented images, while the test-set is the same as before. It is observed that image augmentation provides some improvement to the classification results. Despite the small improvement of the KNN classifier, which only gains 0.6% accuracy on test data, the SVM classifier more than double's its performance. Taking into account the training results (not shown here), this increase in performance is justified by the reduction of overfitting resulting from data augmentation.

4 Conclusion

ImageNet winning networks already achieve an accuracy greater than 95%, but when adapted to classify skin lesions their performance drops to quite modest results, even using data augmentation. This work performed transfer learning to classify skin lesions as malignant or benign using 5 cornerstone neural network architectures that have been proven to produce high results on other domains. The results demonstrate that there is significant room for further research, using highly accurate networks and transfer learning for specific classification in the field of medical imaging. In particular, it is necessary to investigate how to improve transfer learning performance using networks trained on completely different domains.

Acknowledgments

This work was supported by the Fundação para a Ciência e Tecnologia, Portugal, under PhD Grant SFRH/BD/128669/2017 and PlenoISLA project in the scope of R&D Unit 50008, through national funds and where applicable co-funded by FEDER – PT2020 partnership agreement.

References

[1] International skin imaging collaboration: Melanoma project website. `https://isic-archive.com/`. Accessed: 01.12.2017.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[3] Robert J. Friedman, Darrell S. Rigel, and Alfred W. Kopf. Early detection of malignant melanoma: The role of physician examination and self-examination of the skin. *CA: A Cancer Journal for Clinicians*, 35(3):130–151, 1985.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conf. on comp. vision and pattern recognition*, pages 770–778, 2016.

[5] Howard L Kaufman. *The melanoma book: a complete guide to prevention and treatment*. Gotham, 2005.

[6] K. Korotkov and R. Garcia. Computerized analysis of pigmented skin lesions: a review. *Artif. Intel. in Medicine*, 56(2):69–90, 2012.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, 5:5858–5869, 2017.

[9] Sameena Pathan, K Gopalakrishna Prabhu, and PC Siddalingaswamy. Techniques and algorithms for computer aided diagnosis of pigmented skin lesions – a review. *Biomedical Signal Processing and Control*, 39:237–262, 2018.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Paper:#11

Prediction of Healing Deformities After Breast Conserving Surgery

By: Bessa, S. Oliveira, H. Cardoso, J. Oliveira, S. Zolfagharnasab, H.

Prediction of Healing Deformities After Breast Conserving Surgery

Sílvia Bessa[1,2]
silvia.n.bessa@inesctec.pt
Hooshiar Zolfagharnasab[1]
hooshiar.h.z@ieee.org
Sara P. Oliveira[1,3]
sara.i.oliveira@inesctec.pt
Jaime S. Cardoso[1,3]
jaime.cardoso@inesctec.pt
Hélder P. Oliveira[1,2]
helder.f.oliveira@inesctec.pt

[1] INESC TEC
Porto, Portugal
[2] FCUP
Porto, Portugal
[3] FEUP
Porto, Portugal

Abstract

Breast conserving surgery (BCS) has become the preferred method to treat breast cancer for most of the patients. Although it has lower aesthetic impact than mastectomy, the outcome can still be unappealing for most women. The prediction of breast healing deformities caused by BCS is important to both patients and surgeons and can be a determining factor in the decision process. In this work, a methodology based on Random Forests (RF) trained with adaptive weights is proposed to predict breast surface displacements caused by BCS, one year after surgery.

1 Introduction

After BCS, breast tissues adapt to fill the void left by removing the tumor, which typically results in the loss of breast volume and contraction. This healing process takes almost one year to stabilize the breast in its new shape. The final aesthetic outcome can be affected by different surgical practices and expertise as well as some breast specific characteristics such as volume and density, and tumor porperties such as size and location [1]. Therefore, the prediction of breast deformities after cancer surgery is a complex task.

In literature, strategies to model breast deformations are abundant and designed to different applications: estimate pose transformation, assist registration tasks among different radiological exams, model breast deformation, guide surgery or predict the healing process of the breast after tumor removal, among others [2]. These applications have in common the use of biomechanical models to predict deformations, which have inherent limitations to be introduced in clinical practice. Challenges include high computational time and the use of simplistic representations of the breast biomechanics and unverified parameters in most models. [3]. Alternative strategies to model breast deformation encompass the fitting of parametric models, physical equations to describe known breast deformations, user-intuitive parameters to change breast shape, or databases of known cases to simulate breast surgery outcomes [5]. In this work (adapted from [6]), the use of a machine learning strategy to predict breast shape deformations caused by cancer surgery is explored.

2 Dataset

To deal with the lack of an appropriate dataset containing 3D surface data before and after surgery to train a learning model, an in-house semi-synthetic dataset was created, taking advantage of available Magnetic Resonance Image (MRI) data of breast cancer patients with anatomical structures of interest annotated, as detailed in [6]. Due to the lack of data after surgery, the healing process was simulated using the multiscale biomechanical model of breast healing proposed by Vavourakis *et al*. [4]. In detail, 6 breast point clouds (PCLs, obtained from MRI data) were used, taking into account an uniform distribution of breast volumes (2 small, 2 medium and 2 large breasts) and laterality (3 left and 3 right breasts). New dataset instances were created by sequentially defining 4 different breast densities for each PCL, according to BI-RADS® reporting system ($4 \times 6 = 24$ cases), then different quadrants for the tumor location ($4 \times 24 = 96$ cases) and, finally, 3 different tumor sizes for each location ($3 \times 96 = 288$ cases). In the end, the dataset sums up to a total of 288 cases representing all the possible combinations of the most prominent clinical factors reported to affect breast shape after BCS.

3 Methodology

The prediction of healing deformities can be understood as a regression problem that takes as input the coordinates of points (continuous variables in 3D space) and clinical features (see Table 1), and outputs the new coordinates of points after surgery. Alternatively, the problem can be decomposed in the prediction of coordinates displacements, which are subsequently added to the pre-surgery coordinates of points in order to predict the post-surgery locations. This alternative overcomes the necessity to map different breasts to the same reference system, which simplifies the task. Hence, this regression model can be expressed as:

$$\begin{bmatrix} P^{pre} & F \\ \vdots & \vdots \end{bmatrix} \xrightarrow{f} \begin{bmatrix} disp^{pre \to post} \\ \vdots \end{bmatrix}, \tag{1}$$

where P^{pre} is the pre-surgery PCL, F is the feature list per instances (pre-surgery points), f is the regression model to be determined, and $disp^{pre \to post}$ represents the displacements necessary to the obtain the predicted post-surgery PCL (P^{pred}):

$$P^{pre} + disp^{pre \to post} = P^{pred} \tag{2}$$

In this work, each component of the 3D $disp^{pre \to post}$ vector was predicted training individual RF model for each axis. The optimization of the RF parameters and the calculation of the models performance was carried out with a leave-one-patient-out (LOPO) cross-validation strategy, devised to deal with the high correlation of dataset instances derived from the same patient. In detail, for each split, all dataset instances derived from the same patient were used for testing, and the remaining data derived from other patients was used to optimize the parameters of the model. The optimized parameters were the number of estimators (trees), the maximum number of features in each tree, and the leaf size chosen from the ranges $\{5, 10, ..., 500\}$, $\{2, 3, ..., 23\}$, and $\{1, 2, ..., 5\}$, respectively. The effect of the PCLs sampling ($\{0.05, 0.1, ..., 1\}$) was also explored to find the optimal threshold between the training computational cost and the performance of the model.

During training, adaptive weights were assigned to instances and updated at each iteration. The weights varied between $\{1, 2, ..., 6\}$, and were proportional to the distance between predicted and target displacements. The range of the weights was defined using the maximum error of an optimized RF model trained without them, as the maximum value for weights. The adaptive process stopped when a fixed number of iteration was reached (in this case 100), or the objective function (OF) did not change after 3 consecutive iterations. To select the best overall model parametrization, two OFs were explored: average, and Hausdorff (maximum) distances between post-surgery and predicted PCLs.

4 Results

The performance of the proposed regression model was assessed both numerically and visually. Numerical evaluation was accomplished measuring distances between predicted (as the source) and post-surgery PCLs (as the target). Equation 3 denotes the Euclidean *point-wise distance (p2p)*, which uses known point correspondences between PCLs:

$$D^{p2p} = \frac{1}{N} \sum_{i=1}^{N} d(P_i^{source}, P_i^{target}), \tag{3}$$

Features	ID	Type	Space	Description
Point's coordinate	p_x p_y p_z	Quantitative Continuous	$\in \mathbb{R}^3$	Breast points coordinates in 3D space. The center of geometry of PCLs should be translated to the origin.
Coordinate difference to the excised cylinder	$disp_x$ $disp_y$ $disp_z$	Quantitative Continuous	$\in \mathbb{R}^3$	Difference of each healthy point (signed) of pre-surgery PCL to the excised cylinder
Distance to the excised cylinder	$d_{x,y,z}$	Quantitative Continuous	$\in \mathbb{R}$	Euclidean distance of each point of pre-surgery PCL to the excised cylinder
Polar distance to the excised cylinder	ρ ϕ z	Quantitative Continuous	$\in \mathbb{R}^3$	Polar difference of each healthy point (signed) of pre-surgery PCL to the excised cylinder
Tumor Size	s_1 s_2 s_3	Categorical Ordinal	100 010 001	Defines the size of tumor (5%, 7.5%, or 10% of total breast volume)
Breast Laterality	R L	Categorical Nominal	1 0	Indicates the laterality of breast (right or left)
Breast Density	A B C D	Categorical Ordinal	1000 0100 0010 0001	Determines breast density level (A, B, C, or D)
Tumor Region	R_1 R_2 R_3 R_4	Categorical Nominal	1000 0100 0010 0001	Specifies the region of breast with the tumor (UOQ,UIQ, LOQ, or LIQ)

Table 1: Description of features used in the regression model.

N and d denote the number of points and Euclidean distance, respectively, and P_i^{source} is the corresponding point of P_i^{target}. Lower distances mean the regression model predicts the coordinate of each point closer to its expected location. Equation 4 denotes *global distances*. Contrarily to point-wise distance, global distances disregard corresponding points in favour of the closest ones and can be computed in two directions:

$$\begin{cases} D^{global}_{source\rightarrow target} = \frac{1}{N}\sum_{i=1}^{N} \min_j d(P_i^{source}, P_j^{target}) \\ D^{global}_{target\rightarrow source} = \frac{1}{N}\sum_{i=1}^{N} \min_j d(P_i^{target}, P_j^{source}) \end{cases}, \quad (4)$$

P_j is the nearest point to P_i. Closer reported distances mean more similarity between the two PCLs.

Point-wise and global numerical evaluations obtained with average and Hausdorff OFs are reported in Table 2 and Table 3, respectively. To evaluate the magnitude of the reported distances, an extra comparison is performed with the distance between the two comparing PCLs in case no method is applied (meaning that prediction data is exactly equal to the pre-surgery data). This comparison, so-called *baseline evaluation*, is reported in both tables.

	Average-based OF	Hausdorff-based OF	Baseline evaluation
	D^{p2p}	D^{p2p}	D^{p2p}
μ	1.048	1.189	2.206
σ	0.905	0.981	1.920
Max	5.240	4.083	8.410

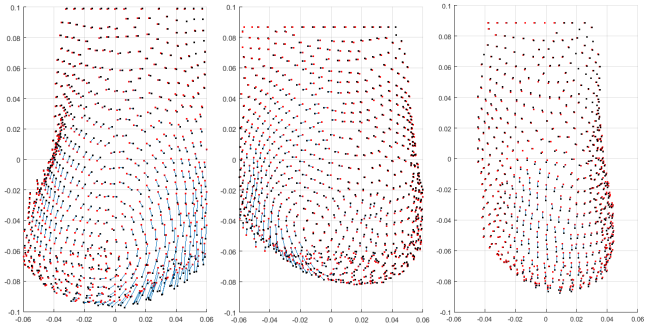
Table 2: Point-wise distance (in *mm*) between predicted and post-surgery PCLs.

	Average-based OF		Hausdorff-based OF		Baseline evaluation	
	$D^{global}_{pred\rightarrow post}$	$D^{global}_{post\rightarrow pred}$	$D^{global}_{pred\rightarrow post}$	$D^{global}_{post\rightarrow pred}$	$D^{global}_{pre\rightarrow post}$	$D^{global}_{post\rightarrow pre}$
μ	0.961	0.951	1.124	1.094	1.758	1.731
σ	0.951	0.861	0.958	0.937	1.333	1.277
Max	5.182	5.178	4.022	3.980	6.512	6.317

Table 3: Global distance (in *mm*) between predicted and post-surgery PCLs.

As expected, average and Hausdorff OF impacted different metrics of performance. When using average OF, the average distance decreased, while using Hausdorff OF decreased the maximum distance at expense of an increased average one. Nevertheless, average distances are around 1*mm* regardless of the OF, while maximum distances are found between 4 and 5 mm. The choice of the OF will then depend on which metric (average or maximum distances) the surgeons value the most.

Besides the reported numerical evaluation, visual comparisons of three predicted breasts are depicted in Figure 1. The depicted predictions have been evaluated with average pair-wise distance of 1.62 *mm* as a poor prediction, 1.044 *mm* as a fair prediction, and 0.827 *mm* as a good prediction.



(a) Poor prediction $D_{pred\rightarrow post} = 1.624$ *mm*. (b) Fair prediction $D_{pred\rightarrow post} = 1.044$ *mm*. (c) Good prediction $D_{pred\rightarrow post} = 0.827$ *mm*.

Figure 1: Visual evaluation: post-surgery and predicted PCLs are shown in red and black, respectively; blue arrows depict displacements between corresponding pair-wise points.

5 Conclusion

In this paper, the use of machine learning techniques to predict healing deformities after BCS has been addressed. A RF model trained with adaptive weights and a LOPO strategy to avoid over-fitting and bias has shown to have promising results, with average distances between predicted and target PCLs in the order of 1 *mm*, suitable for clinical usage. The model was trained using a semi-synthetic dataset generated using MRI data from real patients combined with a multiscale biomechanical model to simulate post-surgical breast shape according to different clinical features.

Acknowledgements

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia as part of project UID/EEA/50014/2013 and within PhD grant number SFRH/BD/-115616/2016.

References

[1] Maria João Cardoso, Hélder Oliveira, and Jaime Cardoso. Assessing cosmetic results after breast conserving surgery. *Journal of surgical oncology*, 110(1):37–44, 2014.

[2] Thiranja P Babarenda Gamage, Vijayaraghavan Rajagopal, Poul MF Nielsen, and Martyn P Nash. Patient-specific modeling of breast biomechanics with applications to breast cancer detection and treatment. In *Patient-Specific Modeling in Tomorrow's Medicine*, pages 379–412. Springer, 2011.

[3] Vijay Rajagopal, Poul MF Nielsen, and Martyn P Nash. Modeling breast biomechanics for multi-modal image analysis - successes and challenges. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 2(3):293–304, 2010.

[4] Vasileios Vavourakis, Bjoern Eiben, John H Hipwell, Norman R Williams, Mo Keshtgar, and David J Hawkes. Multiscale mechano-biological finite element modelling of oncoplastic breast surgery - numerical study towards surgical planning and cosmetic outcome prediction. *PloS one*, 11(7), 2016.

[5] Hooshiar Zolfagharnasab, Jaime S Cardoso, and Hélder P Oliveira. Fitting of breast data using free form deformation technique. In *International Conference Image Analysis and Recognition*, pages 608–615. Springer, 2016.

[6] Hooshiar Zolfagharnasab, Sílvia Bessa, Sara P Oliveira, Pedro Faria, João F Teixeira, Jaime S Cardoso, and Hélder P Oliveira. A regression model for predicting shape deformation after breast conserving surgery. *Sensors*, 18(1):167, 2018.

Paper:#13

Improving ECG-Based Biometric Identification Using End-to-End Convolutional Networks

By: Pinto, J. Cardoso, J. Lourenço, A.

Improving ECG-Based Biometric Identification Using End-to-End Convolutional Networks

João Ribeiro Pinto[1]
jtpinto@fe.up.pt
Jaime S. Cardoso[1]
jaime.cardoso@fe.up.pt
André Lourenço[2]
arl@cardio-id.com

[1] INESC TEC & Faculdade de Engenharia
Universidade do Porto
Porto, Portugal

[2] CardioID Technnologies, Instituto Superior de Engenharia
de Lisboa, & Instituto de Telecomunicações
Lisboa, Portugal

Abstract

Using Convolutional Neural Networks (CNNs), this work studies how the integration of all processes needed for biometric recognition on a single model improves ECG-based subject identification. An end-to-end unidimensional CNN is proposed, which receives raw blindly-segmented ECG signals and outputs an identification, after being optimised, as a whole, during training. The proposed method was evaluated on the UofTDB collection, offering 96.1% identification rate (IDR), and on the PTB database, attaining 98.6% IDR. When compared with implemented state-of-the-art methods and results reported in the literature, the network showed improved performance and enhanced robustness to the increased noise and variability of off-the-person signals, even with larger sets of subjects.

1 Introduction

The electrocardiogram (ECG) is a biosignal that is defying the dominion of face, fingerprint, voice, and iris over research and industry in biometric recognition. The ECG is universal, sufficiently permanent, and easily measurable, with increased comfort if acquired using non-intrusive techniques. Besides this, its hidden nature and inherent liveness information place it as a promising biometric trait.

Research in ECG-based biometrics [6] has been quickly moving from highly clean and controlled medical signals (designated as *on-the-person*) towards more comfortable and realistic settings (*off-the-person)*, using fewer electrodes on the subjects' fingers or palms. With on-the-person signals, Matta *et al*. [5] used linear discriminant analysis and a nearest neighbour classifier with autocorrelation coefficients from ECG segments, while Brás and Pinho [2] used Kolmogorov-based compression on signals denoised using moving average, notch, and lowpass filters. Eduardo *et al*. [3] trained a deep autoencoder to extract signal representations fed to a nearest neighbour classifier, while Salloum and Kuo [8] proposed a Recurrent Neural Network (RNN) with Long Short-Term Memory and Gated-Recurrent Units.

With off-the-person signals, Lourenço *et al*. [4] used average heartbeats, denoised and normalised in time and amplitude. Pinto *et al*. [7] proposed the extraction of DCT coefficients to be used by a Support Vector Machine classifier. Wieclaw *et al*. [10] fed individual segmented heartbeats to a multilayer perceptron (MLP). Zhang *et al*. [11] extracted bidimensional representations the acquired ECG signals and used a 2D Convolutional Neural Network (CNN) for classification. In general, the inferior results reported with off-the-person signals, relative to those with on-the-person signals, illustrate the effect of increased noise and variability on the identification performance.

We observe that, so far, most electrocardiogram-based methods for biometric recognition are composed of several separate processes, each focused and optimised for a single task. Even the recently proposed deep learning algorithms rely on additional techniques for the denoising and feature extraction alongside convolutional or recurrent networks. However, it is reasonable to assume that the separate optimisation of such processes can limit the achievable performance of the recognition methods. Combining all processes in a single network would enable joint, simultaneous, and coordinated optimisation for better performance and robustness to the enhanced noise and variability of off-the-person signals.

Hence, in this work, we study the use of a convolutional neural network (CNN) for ECG-based biometric identification. The proposed network is end-to-end: it receives raw, blindly-segmented electrocardiogram segments and outputs a predicted identity among all enrolled subjects. Thus, it dismisses separate techniques and assumes control over all processes required for robust recognition, expectedly offering improved identification performance and robustness. Two large databases with on-the-person and off-the-person signals were used for performance evaluation. The identification rate (IDR, or accuracy) results were compared with selected state-of-the-art methods and literature results.

2 Proposed Methodology

As aforementioned, the proposed methodology consists of a unidimensional convolutional neural network that receives a raw ECG segment and outputs a decision on the respective identity. The CNN, as shown in Fig. 1 is composed of four convolutional layers, three max-pooling (MaxPool) layers, and one fully-connected (dense) layer.

The convolutional and pooling layers compose the feature-focused part of the model. The first two convolutional layers have 24 filter banks, while the remaining have 36 filter banks, while all filters' size is 1×5. These enable the network to learn to represent the input signal in the most advantageous way for the identification task at hand. Rectified Linear Unit (ReLU) was used as activation for all convolutional layers. The pooling layers, with size 1×5 and placed between each two consecutive convolutional layers, greatly reduce the dimensionality of the feature maps, reducing processing load during both training and inference, effectively making the model more efficient.

Receiving the flattened feature maps from the last convolutional layer, the fully-connected layer is in charge of classification. This layer is composed of N neurons (where N is the total number of enrolled identities) and will, at each neuron, appropriately weigh each input feature to output expected scores for each identity. Softmax activations are used for a normalised distribution of those scores.

As discussed above, the model is optimised as a whole, for the task at hand, in order to maximise the achievable identification performance. The training of the network is performed using the optimiser Adam, based on sparse categorical cross-entropy loss, with empirically tuned learning rate.

During training, both dropout and data augmentation were used to avoid overfitting. Dropout was used between the last convolutional layer and the fully-connected layer. Unidimensional data augmentation was applied to the training signals by dividing each input segment into five 1 second subsegments and randomly shuffling them. The augmentation was implemented using an online data generator.

3 Evaluation Settings

The proposed methodology was evaluated on the UofTDB [9] and PTB [1] collections. UofTDB is a collection of off-the-person ECG signals acquired at 200 Hz using dry electrodes on the fingertips of 1019 subjects, in up to six sessions (over a period of up to six months) on five different postures. It enables the study of the impact of long-term variability and movement/activity noise on the recognition performance. PTB holds 15-lead ECG signals from 290 subjects at rest, acquired at 1000 Hz in medical settings. For this study, we resampled PTB signals to 200 Hz and, as common in the literature, we used solely Lead I.

The signals were divided into five-second segments (1000 samples) to make up the dataset, which was randomly divided 70% for training and 30% for testing. IDR (identification rate, or classification accuracy) was selected as the performance metric. The state-of-the-art methods of Eduardo *et al*. [3] and Matta *et al*. [5] were implemented and tested in the same conditions. Along with the entire UofTDB dataset of 1019 subjects, two subdatasets with 25 and 100 subjects were used to study performance in smaller populations.

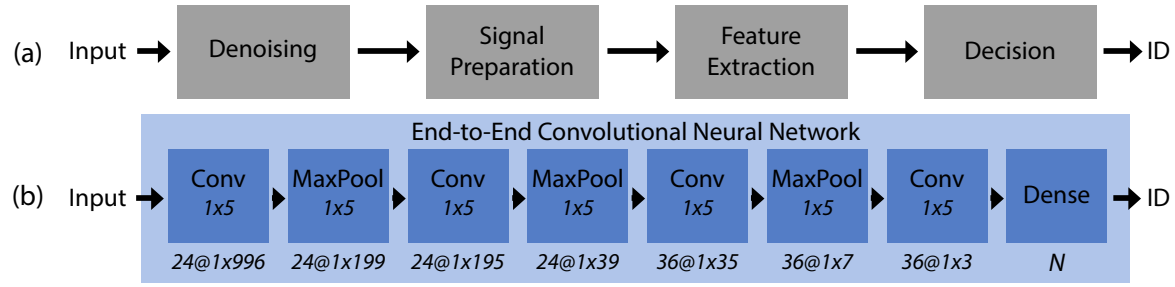


Figure 1: The conventional structure of an ECG-based biometric identification algorithm (a), compared with the proposed end-to-end network (b).

Table 1: Performance evaluation results of the proposed method and the implemented state-of-the-art approaches.

	IDR per database and number of subjects			
	UofTDB			PTB
Method	25	100	1019	290
Proposed Method	99.7%	98.7%	96.1%	98.6%
AC/LDA [5]	96.8%	95.5%	90.0%	98.8%
Autoencoder [3]	97.0%	93.6%	85.0%	99.5%

Table 2: Comparison between the proposed method's performance and the results reported in the literature (N.S. - number of subjects in the database; O.P. - off-the-person).

Method	Database and N.S.	O.P.	IDR
Proposed Method	UofTDB - 1019	Yes	96.1%
Wieclaw *et al.* [10]	Private - 18	Yes	89.0%
Lourenço *et al.* [4]	Private - 16	Yes	94.3%
Zhang *et al.* [11]	Several - 10	Yes	98.4%
Pinto *et al.* [7]	Private - 6	Yes	94.9%
Proposed Method	PTB - 290	No	98.6%
Salloum and Kuo [8]	ECG-ID - 90	No	100%
Brás and Pinho [2]	PTB - 52	No	99.9%

4 Results and Discussion

The performance results for the evaluated methods are presented in Table 1. The proposed network achieved 96.1% when evaluated with the entire UofTDB dataset of 1019 subjects and 98.6% on the PTB dataset. These are, arguably, desirable results for an ECG-based identification system, considering the acquisition settings and the large sets of subjects.

With the on-the-person PTB dataset, the best results were attained by the Autoencoder method proposed by Eduardo *et al.* [3]. However, with the more challenging UofTDB datasets, the proposed method outperformed both state-of-the-art methods in all cases, showing improved robustness to the increased noise and variability carried by off-the-person signals. With these datasets, the IDR difference between the proposed method and the best state-of-the-art method was 2.9% for 25 subjects, but increases to 6.1% for 1019 subjects, showcasing better scalability of the proposed network to larger sets of identities.

When compared with other performance results recently reported in the literature, the proposed method shows promise. Despite the different evaluation settings and much larger number of identities on the evaluation datasets, the proposed method outperformed all literature methods evaluated on off-the-person ECG signals (see Table 2) except the one proposed by Zhang *et al.* [11] and, with on-the-person signals (PTB), it was able to nearly match the almost perfect performance reported on the literature.

5 Conclusion and Future Work

This work focuses on the study of an end-to-end convolutional neural network for biometric identification based on minimally obtrusive, off-the-person electrocardiogram signals. The proposed network was able to adequately integrate all processes needed for recognition in a single model that receives raw signals and outputs predicted identities.

Evaluated with large ECG signal collections, the proposed method was successful in outperforming other state-of-the-art methods with signals acquired in off-the-person settings, and offered competitive performance when compared with several on-the-person results reported in the literature. The network showed improved identification performance, even with larger sets of subjects and considerably increased noise and variability of off-the-person signals, ultimately showing promise for successful implementation in real biometric applications.

Acknowledgements

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project "POCI-01-0145-FEDER-006961", and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, as part of project "UID/EEA/50014/2013", and within the PhD grant number "SFRH/BD/137720/2018".

The authors wish to express their gratitude to the BioSec.Lab of the University of Toronto for granting access to the UofTDB database, and to acknowledge Physionet and the creators of the PTB database.

References

[1] R. Bousseljot, Kreiseler D., and A. Schnabel. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. *Biomedizinische Technik*, 40(s1):317–318, 1995.

[2] S. Brás and A. J. Pinho. ECG biometric identification: A compression based approach. In *37th EMBC*, pages 5838—-5841, 2015.

[3] A. Eduardo, H. Aidos, and A. Fred. ECG-based Biometrics using a Deep Autoencoder for Feature Learning: An Empirical Study on Transferability. In *ICPRAM 2017*, pages 463–470, 2017.

[4] A. Lourenço, H. Silva, and A Fred. Unveiling the Biometric Potential of Finger-based ECG Signals. *Computational Intelligence and Neuroscience*, 2011:720971, 2011.

[5] R. Matta, J. Lau, F. Agrafioti, and D. Hatzinakos. Real-time continuous identification system using ECG signals. In *24th CCECE*, pages 1313 – 1316, 2011.

[6] J. R. Pinto, J. S. Cardoso, and A. Lourenço. Evolution, Current Challenges, and Future Possibilities in ECG Biometrics. *IEEE Access*, 6:34746–34776, 2018.

[7] J. R. Pinto, J. S. Cardoso, A. Lourenço, and C. Carreiras. Towards a Continuous Biometric System Based on ECG Signals Acquired on the Steering Wheel. *Sensors*, 17(10):2228, 2017.

[8] R. Salloum and C. C. J. Kuo. ECG-based biometrics using recurrent neural networks. In *ICASSP 2017*, pages 2062–2066, 2017. doi: 10.1109/ICASSP.2017.7952519.

[9] S. Wahabi, S. Pouryayevali, S. Hari, and D Hatzinakos. On evaluating ECG biometric systems: session-dependence and body posture. *IEEE TIFS*, 9(11):2002–2013, 2014.

[10] L. Wieclaw, Y. Khoma, P. Falat, D. Sabodashko, and V. Herasymenko. Biometric identification from raw ECG signal using deep learning techniques. In *9th IEEE IDAACS*, pages 129–133, 2017.

[11] Q. Zhang, D. Zhou, and X. Zeng. Pulseprint: Single-arm-ECG biometric human identification using deep learning. In *2017 IEEE UEMCON*, pages 452–456, 2017.

Paper:#15

Rotation Equivariant Convolutional Layers in Deep Neural Networks

By: Castro, E. Pereira, J. Cardoso, J.

Rotation Equivariant Convolutional Layers in Deep Neural Networks

Eduardo Castro
emcastro@inesctec.pt
José Costa Pereira
jose.c.pereira@inesctec.pt
Jaime S. Cardoso
jaime.cardoso@inesctec.pt

Centro de Telecomunicações e Multimédia
INESC TEC
Porto, Portugal

Abstract

One of the key ideas in the design of Convolutional Neural Networks is the parameter sharing property of its convolutional layers. Due to this property, the response of a convolutional layer is equivariant to input translations which is a good prior in most image recognition tasks. Other types of equivariance priors can be encoded in the architecture of CNNs. In this work we extend the parameter sharing property to accommodate for rotations of the input. We show that this prior can lead to better generalization when encoded in the early layers of CNNs even in tasks where the input data is not symmetric to rotation.

1 Introduction

Convolutional Neural Networks (CNNs) are deep feed-forward neural networks designed to process data in the form of multiple arrays [8]. They work by applying a sequence of simple parametric functions to the input. Through gradient descent algorithms, CNNs can be optimized to learn hierarchical representations which have been shown useful for many practical applications including image classification [5].

One of the key ideas behind the design of CNNs is the parameter sharing property. In each convolutional layer, a filter-bank is used to filter the image. In this operation, each filter is applied to all regions of the input. Intuitively, this seems a good prior. If a motif can appear in one part of the image it is likely that it will appear in other locations as well. This is particularly evident for early layers in a model. For instance, if we consider the problem of face recognition using frontal photos of the face, low level features, such as edges, are common across the image, while higher level ones, such as mouths, appear in a specific location. As a necessary consequence of the parameter sharing property, the response of convolutional layers is translation equivariant if we disregard edge effects: a translation in the input will cause a translation in the output.

This parameter sharing property and consequently the equivariant response can be extended to other transformations. Cohen and Welling [3] proposed a generalization of the convolutional layer to account for equivariances to transformations different from translation. In this work we apply this method for rotation transformations. We show how to create models which are more robust to input rotations by making use of the parameter sharing property of CNNs. Our experimental results show that this technique can be useful not only in rotation invariant problems but also in problems which do not exhibit this characteristic, particularly if a small amount of training data is available.

2 Related Work

If we consider rotation invariant image classification problems, those in which a rotated input can in theory appear in the data, robustness to orientation is desirable as it often leads to better generalization. The most popular way to obtain this is by simply using data augmentation [1]. A more elaborate method proposed by Cheng *et al*. [2] works by not only providing transformed examples to the model but also penalizing different representations between the original and transformed input in the loss function.

Some authors obtain robustness not by rotating the training data but rather by model design. By creating models designed specifically to deal with known symmetries in the training data we are able to reduce the search space during optimization which can lead to improvements in convergence speed and generalization. Architectural changes often fall in one of two categories: i) regular Group Equivariant Convolutional Networks (G-CNNs) and ii) steerable G-CNNs. We are interested in the first group but we point to [4] for a more complete explanation. Regular G-CNNs work by making use of the group equivariant convolutional layer [3]. This module works by tying the weights of the filter-bank in such a way that the output becomes equivariant under a specific group of transformations. Additionally, from an equivariant model it is easy to obtain an invariant one. Examples of works that apply the group equivariant convolutional layer to rotation include [9] and [3]. While Marcos *et al*. [9] used a one-layer-CNN rotation equivariant layer to classify texture, Cohen and Welling [3] generalized this concept to finite groups of transformations in CNNs with more than one convolutional layer. Both works showed better accuracy in rotation invariant image recognition problems.

3 Methods

3.1 Equivariance and Invariance

Considering any group of transformations G, applied to the input I, we say $f(.)$ is invariant under group G if Eq. 1 holds. We say instead that $f(.)$ is equivariant under group G if for each element of $g \in G$ there exists an h so that Eq. 2 holds.

$$f(I) = f(g.I) \tag{1}$$
$$h.f(I) = f(g.I) \tag{2}$$

In this work we consider the group of transformations composed of plane rotations of $\{0^o, 90^o, 180^o, 270^o\}$. We call this group $p4$ in accordance with the work of [3]. One important notion is that for image recognition problems rotation equivariance is more useful than invariance for two reasons: i) equivariance keeps information about the orientation of the detected motif; ii) an equivariance response can be made invariant by combining the responses in all orientations. Note that because the response is equivariant we do not need to compute $f(g.I)$ for all transformations but only $h.f(I)$, which is typically much cheaper.

3.2 Rotation Equivariant Convolutional Layer (RECL)

In this section we will denote w and I^k as the convolutional weights and the output after layer k, respectively. The input image is considered the feature map with $k = 0$. We will refer to the well-known convolution operation in convolutional layers as $conv2d(w, I)$. We consider the group of transformations $p4$ with the following transformations $\{T_0, T_{90}, T_{180}, T_{270}\}$.

One way to obtain the response for a rotated input is by rotating the weights, filtering the image and rotating the result in the opposite direction:

$$T_i \circ conv2d(T_{-i} \circ w, I) = conv2d(w, T_i \circ I) \tag{3}$$

If we consider the input I^0, we can define the output of the first convolutional layer as the stack of responses obtained by filtering the image with multiple rotations of the original filter-bank:

$$I^1 = [I^1_0; I^1_{90}; I^1_{180}; I^1_{270}] \tag{4}$$
$$I^1_j = conv2d(T_j \circ w, I^0) \tag{5}$$

Consider the output $I^{1\prime}$ obtained by performing the same operation but on $T_{90} \circ I^0$. By making use of eq. 3 $I^{1\prime}_j$ is given by:

$$I^{1\prime}_j = conv2d(T_j \circ w, T_{90} \circ I^0) = T_{90} \circ conv2d(T_{j-90} \circ w, I^0) \tag{6}$$

Table 1: Test accuracy for transformed MNIST for multiple models. The first line indicates the number of images used for training.

Model	100	1k	10k	55k
Baseline	71.5%	93.2%	**98.5%**	**99.2%**
Rot-2	72.0%	93.1%	98.4%	**99.2%**
Rot-4	**73.0%**	**93.7%**	**98.5%**	**99.2%**
Rot-5	72.1%	92.9%	98.3%	**99.2%**
Baseline max-out	72.0%	93.8%	98.6%	99.2%
RICNN	**73.9%**	**94.3%**	**98.8%**	**99.5%**

We obtain the following relationship between the responses of the original and rotated input:

$$I^1 = [I^1_0; I^1_{90}; I^1_{180}; I^1_{270}] \tag{7}$$

$$I^{1\prime} = T_{90} \circ [I^1_{270}; I^1_0; I^1_{90}; I^1_{180}] \tag{8}$$

$$\tag{9}$$

By rotating the output and shifting it in the channels dimensions we are able to obtain the response for a transformed input, satisfying the definition of equivariance. Note that a rotation of I^0 not only rotates I^1 but also shifts it in the channels dimensions. Due to this, for the second layer, the transformation of the weights should accommodate for this. I^2 is therefore given by:

$$I^2 = [I^2_0; I^2_{90}; I^2_{180}; I^2 1_{270}] \tag{10}$$

$$I^2_j = conv2d(S_j \circ (T_j \circ w), I^0) \tag{11}$$

Where S_j is a circular shift operation that acts in the channels dimensions, and shifts the array $\frac{j}{90} \times \frac{n}{4}$, with n being the number of channels in the input (I^1). Due to lack of space we will not prove the equivariance of the second layer but it is straightforward if we consider that the response to a shifted input can be obtained by filtering the original input with the filter-bank shifted in the opposite direction and shifting back the result:

$$S_i \circ conv2d(S_{-i} \circ w, I) = conv2d(w, S_i \circ I) \tag{12}$$

In terms of implementation, present-day deep learning frameworks have routines which allow shifting and rotating multidimensional arrays. As such, implementing the RECL can be done by simply manipulating the weights.

4 Experimental Section

4.1 MNIST

The well known MNIST dataset [7] contains images of handwritten digits with size 28×28. We expand this dataset by rotating each image by 90^o, 180^o and 270^o. We do not use intermediate rotations as we are only interested in creating an artificial problem invariant to $p4$ transformations. The baseline CNN architecture is composed of two blocks, each with two convolutional and one max-pooling layers, and two fully connected layers before the output. After each linear layer (convolutional or fully-connected) a ReLU activation was used. By substituting the n-first convolutional layers by their rotation equivariant counterpart, while keeping the same number of channels, we obtain additional models, called Rot-n to compare against our baseline.

The results, depicted in table 1, show that the introduction of equivariant layers leads to better generalization. This effect increases as the size of the training dataset decreases. This is to be expected as the effect of overfitting (lack of generalization) is more notorious when a small number of examples is available. The Rot-5 model performs worse than Rot-4. In this model, the first fully connected layer is implemented as a convolutional one with the weight structure described in the previous section. The drop in accuracy shows that the rotation equivariance prior is no longer useful at that level.

In the same dataset we created a rotation invariant CNN by max-pooling features from different orientations in the Rot-5 model. We compare this against a model with 4-way maxout activation, which effectively performs the same operation but without a structured equivariant representation. The results, depicted in table 1, show better generalization for the global invariant model.

Table 2: Test Accuracy for the downsampled SmallNORB dataset.

Baseline	Rot2	Rot4	Rot7	Rot8
93.2%	**94.0%**	93.8%	91.2%	90.8%

Table 3: Test Accuracy for the SmallNORB dataset.

Baseline	Rot2	Rot4	Rot7	Rot10	Rot11
93.0%	93.8%	93.9%	93.9%	**94.2%**	90.6%

4.2 SmallNORB dataset

The Small Norb dataset [6] is composed of photos of 50 toys equally divided in 5 categories under different lightning conditions, elevations and azimuths, with no color or background. Note that in this case, the image classification problem is not invariant under rotation. During training, random crops with side length equal to $\frac{2}{3}$ of the image are taken. For testing only the central crop is evaluated. The experiment was run twice, first with the images downsampled by a factor of two and then with the original size. As with the previous experiment, models were created by sequentially substituting normal convolutions by the rotation equivariant version. The test accuracy obtained during training by each model is shown in table 2 for the downsampled and 3 for the original datasets.

As we gradually introduce rotation equivariant convolutional layers in the base architecture the validation accuracy increases initially, achieves a maximum and then starts decreasing. Additionally, for images with more resolution the ideal number of RECL increases. This suggests that the rotation equivariance prior can be useful in early layers even for problems which are not invariant to rotation. Intuitively, this is to be expected: it is probable that low level features (eg: edges) appear in multiple orientations while higher level ones (eg: chairs) will almost always appear with the same orientation.

5 Conclusion

The introduction of adequate priors in the architecture of CNNs reduces the search space in optimization which can lead to faster convergence and better generalization. With this work we have shown that equivariance to rotation can be useful in the early layers of CNNs even for image recognition problems that do not present this symmetry in the data at a global level. Future research will focus on finding ways of encouraging the equivariace prior, instead of forcing. The additional flexibility of such an approach should be useful at dealing with problems without known symmetries.

Acknowledgements This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia as part of project «UID/EEA/50014/2013» and within PhD grant number «SFRH/BD/136274/2018».

References

[1] E. Castro, J. S. Cardoso, and J. C. Pereira. Elastic deformations for data augmentation in breast cancer mass detection. In *2018 IEEE EMBS International Conference on Biomedical Health Informatics*.

[2] G. Cheng, P. Zhou, and J. Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, Dec 2016.

[3] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016.

[4] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *ICML*, 2018.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

[6] Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR 2004*.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[8] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

[9] D. Marcos, M. Volpi, and D. Tuia. Learning rotation invariant convolutional filters for texture classification. In *ICPR*, 2016.

Paper:#16

Radio-Pathomics Approach for Breast Tumor Signature: an overview

By: Oliveira, S. Cardoso, M. Cardoso, J. Oliveira, H.

Radio-Pathomics Approach for Breast Tumor Signature: an overview

Sara P. Oliveira[1,2]
sara.i.oliveira@inesctec.pt

Jaime S. Cardoso[1,2]
jaime.cardoso@inesctec.pt

Maria J. Cardoso[1,3]
maria.joao.cardoso@fundacaochampalimaud.pt

Hélder P. Oliveira[1,4]
hfpo@inesctec.pt

[1]INESC TEC
Porto, Portugal

[2]Faculty of Engineering, University of Porto
Porto, Portugal

[3]Breast Unit, Champalimaud Foundation
Lisboa, Portugal

[4]Faculty of Sciences, University of Porto
Porto, Portugal

Abstract

The use of medical imaging to support clinical diagnosis in breast cancer is well-established. However, the fully characterization of the tumor is still challenging due to the partial information that each image modality provides. When radiological image-based quantitative features - extracted in approaches known as radiomics - are combined with clinical reports data and features from other image modalities, they can empower clinical decision support models. In particular, since histopathology images are used to characterize the tumor subtype, the combination of radiological and histopathology information has the potential to enable a more detailed diagnose. In this paper the common pipeline of radiomics and pathomics research is presented, as well as some examples of approaches to tackle each task and the main aspects to take into account.

1 Introduction

Breast Cancer is the most common diagnosed cancer and the leading cause of cancer-related deaths, among women, worldwide. During the most recent years, despite its incidence trends have increased, the mortality rate has significantly decreased, due to an earlier detection and better treatment strategies. In fact, the success of treatment depends a lot on a diagnosis in its early stages, that can be access through radiological imaging, such as Mammography (MG), 2D/3D Ultrasound (US) and Magnetic Resonance Imaging (MRI), even before symptoms have developed [1]. Besides imaging, the assessment of histopathological characteristics can determine the cancer sub-type and grade, information of utmost importance for disease prognosis and treatment decision [6].

Beyond its essential role as a diagnostic tool, medical imaging has evolved as a relevant tool for personalized precision medicine, through a promising approach, known as radiomics. These methods rely on the conversion of digital radiological images into high-dimensional quantitative data, working as sources of information reflecting tumor behavior. Thus, imaging and clinical information can be harnessed through computer-learning methodologies, empowering clinical decision support models [7, 9]. A similar approach, known as pathomics, is also applied to histopathology, trying to characterize large volumes of quantitative image-based features extracted from tissue digital images (Whole Sliding Images – WSI) [14].

The main goal of this paper is to provide a global insigth of the radiomics and pathomics research pipeline, highlighting the main aspects to consider in each task.

2 Radiomics & Pathomics

The common radiomics pipeline, that is identical to pathomics, can be summarized in three steps: (a) image acquisition and segmentation, (b) radiomic/pathomic signatures development and (c) statistical analysis. Segmentation of images into a Region or Volume of Interest (ROI/VOI), such as tumorous cells or lesions, is a crucial step for subsequent analyses, as it determines which pixels/voxels within an image are analyzed. Thus, segmentation should be as automatic as possible, providing accurate and reproducible boundaries, avoiding significant variability of a manual process. Once ROIs/VOIs are defined, several imaging features can be extracted to describe tumor phenotype and its relationship with the surrounding tissues. These features are often too many and may not be useful for a particular task. Therefore, methods for task-specific feature dimensionality reduction and feature selection are used. The final step is machine learning-based data analysis, to identify reliable and reproducible findings with potential to be employed within a clinical context. Machine learning approaches, such as decision trees, random forests, support vector machines and, more recently, deep neural networks, have been shown to be promising in knowledge discovery [7, 9, 16].

Over the past few years, these increasingly popular approaches in cancer imaging research, have been applied to different cancer types, e.g. lung cancer, head and neck cancers, or breast cancer [13], demonstrating the capability of radiomics and pathomics to improve cancer detection, diagnosis, choice of therapeutics, staging and prognosis inference, prediction of treatment response, and monitoring [9]. However, they were always used as parallel methodologies for tumor characterization and not towards a multimodal analysis to investigate possible correlations between the information extracted by both approaches.

2.1 Image Acquisition & Segmentation

Currently, in clinical practice, the most common radiological imaging modalities used for breast cancer diagnosis are MG, US and MRI (Figures 1.a, 1.b and 1.c, respectively) [1]. Moreover, the histopathology images (WSI) (Figure 1.d) are used to evaluate biomarkers and identify molecular subtypes.

In order to extract information from images, the first task is to process radiological images [5, 8, 10] and WSI [11], to delineate tumor regions and the surrounding environment. Since radiomics features are used to describe tumor phenotypes, a precise and robust tumor delineation is needed. Moreover, for WSI analysis, features such as, number of cells or nucleus shape, need to be assessed and thus previously segmented. Therefore, efficient, accurate, robust and automatic segmentation methodologies should be implemented, either using traditional image processing approaches, such as spatial fuzzy C-means [5], active contours [8] or, more recently, deep learning techniques, such as encoder-decoder networks [11].

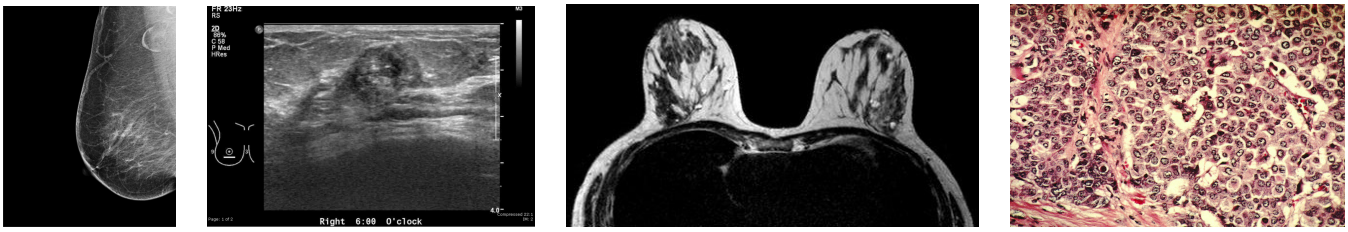
This task is particularly challenging due to subtlety, high resolution and heterogeneity. In fact, lesions indicative of breast cancer are very subtle, making them harder to detect; images size is relatively big requiring bigger models and, in turn, more training data; and image modalities are quite different which requires different processing methodologies to be implemented.

2.2 Feature Extraction

After segmenting the tumor region, the focus is the automatic extraction of its characteristics, such as morphological, texture and dynamic features [5] that can be used as breast cancer image-based biomarkers, especially for breast cancer subtype classification. Apart from looking for features of each modality separately, the use of deep learning techniques to simultaneously describe both particular and shared modality features [2] can be interesting, in order to enhance the combination of information for the tumor characterization at different biological scales.

2.3 Tumour Imaged-based Signature

Finally, the main purpose of this step is to combine the biomedical image-based features previously selected with clinical reports data, and develop a



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

Figure 1: Most common imaging modalities used for breast cancer diagnosis: (a) Mammography - MG, (b) Ultrasound - US, (c) Magnetic Resonance Imaging - MRI and (d) Histopathology images (WSI). Adapted from [3] and [4], respectively.

robust and accurate classification/prediction model to improve breast cancer diagnosis, by creating a complete clinical profile of the patient. These models can be based on classic pattern recognition methodologies, such as SVM's [8, 15] or Naïve Bayes classifiers, or deep learning approaches.

Here, the development of learning models with capability to explain the output results, by indicating the factors used on the decision [17], has major relevance, since interpretability, and not only performance, is important in health care applications [12].

3 Conclusions

Biomedical imaging plays an important role in breast cancer detection and diagnosis and, when image features are combined with clinical reports data, they can empower clinical decision support models. Moreover, assessment of histopathological characteristics is essential to determine the breast cancer subtype, which is of most importance for treatment planning and disease prognosis. Thus, a multimodal image analysis, combining radiological and histopathological data, extracted by radiomic and pathomic approaches, respectively, can provide relevant information for a better insight of each breast cancer clinical case. Over the past few years, these approaches have gained an increasing popularity in cancer imaging research, but were always used as parallel methodologies for tumor characterization and not towards a multimodal analysis to investigate possible correlations between the information extracted by both approaches. This blank space on research can leverage the tumor characterization at different biological scales, towards a better knowledge of tumor biology and improved diagnosis/prognosis.

Acknowledgements This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia as part of project «UID/EEA/50014/2013» and within PhD grant number «SFRH/BD/139108/2018».

References

[1] American Cancer Society Inc. Breast Cancer Facts & Figures 2017-2018, 2017.

[2] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 343–351, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9.

[3] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, Lawrence Tarbox, and Fred Prior. The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository, 2013.

[4] Dr. Cecil Fox via Wikimedia Commons. Breast cancer cells, 10-09-2018. URL `https://commons.wikimedia.org/wiki/File:Breast_cancer_cells_(1).jpg`.

[5] Ming Fan, Hui Li, Shijian Wang, Bin Zheng, Juan Zhang, and Lihua Li. Radiomic analysis reveals DCE-MRI features for prediction of molecular subtypes of breast cancer. *PLoS ONE*, 12(2):1–15, 2017. ISSN 19326203. doi: 10.1371/journal.pone.0171683.

[6] Ziba Gandomkar, Patrick Brennan, and Claudia Mello-Thoms. Computer-based image analysis in breast pathology. *Journal of Pathology Informatics*, 7(1):43, 2016. ISSN 2153-3539. doi: 10.4103/2153-3539.192814.

[7] Robert J Gillies, Paul E Kinahan, and Hedvig Hricak. Radiomics: Images Are More than Pictures, They Are Data. *Radiology*, 278(2): 563–577, 2016. ISSN 0033-8419. doi: 10.1148/radiol.2015151169.

[8] Yi Guo, Yuzhou Hu, Mengyun Qiao, Yuanyuan Wang, Jinhua Yu, Jiawei Li, and Cai Chang. Radiomics Analysis on Ultrasound for Prediction of Biologic Behavior in Breast Invasive Ductal Carcinoma. *Clinical Breast Cancer*, 2017. ISSN 19380666. doi: 10.1016/j.clbc.2017.08.002.

[9] E. J. Limkin, R. Sun, L. Dercle, E. I. Zacharaki, C. Robert, S. Reuzé, A. Schernberg, N. Paragios, E. Deutsch, and C. Ferté. Promises and challenges for the implementation of computational medical imaging (radiomics) in oncology. *Annals of Oncology*, 28(6):1191–1206, 2017. ISSN 15698041. doi: 10.1093/annonc/mdx034.

[10] Wenjuan Ma, Yumei Zhao, Yu Ji, Xinpeng Guo, Xiqi Jian, Peifang Liu, and Shandong Wu. Breast Cancer Molecular Subtype Prediction by Mammographic Radiomic Features. *Academic Radiology*, pages 1–6, 2018. ISSN 18784046. doi: 10.1016/j.acra.2018.01.023.

[11] Sachin Mehta, Ezgi Mercan, Jamen Bartlett, Donald Weaver, Joann Elmore, and Linda Shapiro. Learning to segment breast biopsy whole slide images. *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018-Janua:663–672, 2018. doi: 10.1109/WACV.2018.00078.

[12] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T. Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, pages 1–11, 2017. ISSN 1467-5463. doi: 10.1093/bib/bbx044.

[13] Vishwa Parekh and Michael A Jacobs. Radiomics: a new application from established techniques. *Expert Review of Precision Medicine and Drug Development*, 1(2):207–226, 2016. ISSN 2380-8993. doi: 10.1080/23808993.2016.1164013.

[14] Joel Saltz, Jonas Almeida, Yi Gao, Ashish Sharma, Erich Bremer, Tammy DiPrima, Mary Saltz, Jayashree Kalpathy-Cramer, and Tahsin Kurc. Towards Generation, Management, and Exploration of Combined Radiomics and Pathomics Datasets for Cancer Research. *AMIA Joint Summits on Translational Science proceedings.*, 2017: 85–94, 2017. ISSN 2153-4063.

[15] Elizabeth J. Sutton, Brittany Z. Dashevsky, Jung Hun Oh, Harini Veeraraghavan, Aditya P. Apte, Sunitha B. Thakur, Elizabeth A. Morris, and Joseph O. Deasy. Breast cancer molecular subtype classifier that incorporates MRI features. *Journal of Magnetic Resonance Imaging*, 44(1):122–129, 2016. ISSN 15222586. doi: 10.1002/jmri.25119.

[16] Jie Tian, Di Dong, Zhenyu Liu, Yali Zang, Jingwei Wei, Jiangdian Song, Wei Mu, Shuo Wang, and Mu Zhou. *Radiomics in Medical Imaging—Detection, Extraction and Segmentation*, pages 267–333. Springer International Publishing, 2018. ISBN 978-3-319-68843-5. doi: 10.1007/978-3-319-68843-5_11.

[17] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10590-1.

Paper:#21

On modifying the temporal modeling of HSMMs for pediatric heart sound segmentation

By: Oliveira, J. Renna, F. Mantadelis, T. Gomes, P. Coimbra, M.

On modifying the temporal modeling of HSMMs for pediatric heart sound segmentation

Jorge Oliveira[1]
oliveira_jorge@dcc.fc.up.pt
Francesco Renna[1]
frarenna@dcc.fc.up.pt
Theofrastos Mantadelis[2]
theo.mantadelis@dcc.fc.up.pt
Pedro Gomes[1]
ptmgomes@dcc.fc.up.pt
Miguel Coimbra[1]
mcoimbra@dcc.fc.up.pt

[1] Instituto de Telecomunicações,
Faculdade de Ciências da Universidade do Porto
[2] CRACS & INESC TEC,
Faculdade de Ciências da Universidade do Porto

Abstract

In this paper, we use a real life dataset in order to compare the performance of a hidden Markov model and several hidden semi Markov models that used the Poisson, Gaussian, Gamma distributions, as well as a non-parametric probability mass function to model the sojourn time. Using a subject dependent approach, a model that uses the Poisson distribution as an approximation for the sojourn time is shown to outperform all other models.

1 Introduction

The phonocardiogram (PCG) signal is recorded during an auscultation using an electronic stethoscope. The PCG contains important information concerning the mechanical activity of the heart valves [5]. Signal processing of a PCG has two main goals: the first one is to split the PCG into heart cycles. Each heart cycle is mainly composed by the first heart sound (*S*1), the systolic period (*siSys*), the second heart sound (*S*2), and the diastolic period (*siDia*). The second goal is the detection of other sounds such as the third and fourth heart sounds (*S*3 and *S*4 respectively), heart murmurs, snaps, etc. The methods used for heart sound segmentation can be divided depending on which domain they are applied: the time domain (Shannon energy [6]), the frequency domain (homomorphic filter [4]), etc. Recently, HMMs have shown to be very effective in modeling the heart sound signals: in Gill et al. [3], the signal is pre-processed and a subset of candidates (peaks) are extracted from the homomorphic envelogram, and these candidates are classified using a discrete-time HMM, where the state-distribution is modeled using the time-duration from the preceding candidate to the current one. Schmidt et al. [9] implemented a hidden semi Markov model (HSMM) using the homomorphic filtering envelogram as an observation to the system. This has the advantage (compared to the traditional HMM) that every state duration is explicitly modeled in the state transition matrix. Springer et al. [10] expanded Schmidt's algorithm mainly on the study of the emission probability distribution. He explored a wider range of features and machine learning approaches to model the emission probabilities. This work proposes to enhance the performance of PCG segmentation for pediatric subjects via the following contributions: 1) the study of different distributions for approximating the sojourn time in HSMMs; 2) the experimental validation of the performance of each presented model over a real-life pediatric dataset.

2 Modeling Heart Sounds

2.1 Hidden Markov Models

HMMs are probabilistic models, where the observation sequence $X = x_1, x_2, \cdots, x_n$ depends on the hidden state sequence $S = s_1, s_2, \cdots, s_n$ and the unobserved Markov process [1]. A homogeneous hidden Markov model assumes that the state transition probability matrix Γ is constant over time. In this work, the emission probability distribution is assumed to be a continuous Gaussian function.

2.2 Hidden Semi Markov Models

In a HSMMs, we need also to define, D as the sojourn time distribution matrix. The entries of D are $d_{s_k}(u_k)$ which is the probability of spending u_k units of time in the state $s_k \in \mathcal{S} = \{S1, siSys, S2, siDia\}$. We use five different approaches to model the sojourn time. Four of them are represented by parametric distributions (Geometric, Poisson, Gaussian, Gamma), whereas the last one is a non-parametric probability mass function:

2.3 Initializing the parameters of HMM and HSMM

We use a subject dependent approach, meaning that we train and test with mutually exclusive heart beats of each subject. Parameters are initialized in the training phase, using annotated samples from a given subject. During the testing phase we further optimize our parameters by using different non-annotated samples from the subject. We used exhaustive cross-validation from 1 to 7 training heart beats but we constrained our training sets to those that produce continuous test set. For example, if we train with heart beats $1, 2, 8$ we test with heart beats $3, 4, 5, 6, 7$. For both HMMs and HSMMs, the initial states distribution (π_1) are initialized with equal starting probabilities. The Γ parameters are fixed because in a normal cardiac system the state sequence $\{S1 \rightarrow siSys \rightarrow S2 \rightarrow siDia \rightarrow S1\}$ is fixed. To initialize B we use the annotated samples and compute the parameters $\mu_s, \sigma_s \forall s \in S$ by using the corresponding maximum likelihood estimators. To compute the initial parameters D, we use the annotated time lapse between the beginning and the end of the corresponding state s_k.

2.4 Optimizing the HSMM parameters

The parameters Θ are tuned using the expectation maximization (EM) method [1]. We use the Viterbi algorithm [2] to determine the hidden state sequences corresponding to heart beat components. We recall that the Viterbi algorithm does not attempt to classify every observation sample separately, but instead, it returns the hidden state sequence that maximizes the likelihood function for HMM and HSMM.

3 Methology

3.1 Materials

The DigiScope dataset is composed of samples from 29 different healthy individuals, ranging in age from six months to 17 years old. The recordings have a minimum, maximum and average duration of $\approx 2, 20$ and 8 seconds, respectively. This is a very challenging dataset given the highly varying heart rates of individuals in this age range. Heart sounds have been collected in Real Hospital Português (Recife, Brasil) using a Littmann 3200 stethoscope embedded with the DigiScope Collector [8] technology, recorded at 4000 Hz. The heart sounds have all been collected from the mitral spot. These sounds were then manually annotated by cardiopulmonologists.

3.2 Pre-processing

Following previous literature [3, 4, 7], the system first normalizes and scales the signal to the $[0, 1]$ range. The scaled signal is filtered using a Butterworth lowpass filter of order 10 with a cutoff frequency of 100 Hz, since the majority of the frequency content of the *S*1 and *S*2 (for the DigiScope dataset) is contained in the range $30 - 80$ Hz, as it shown in

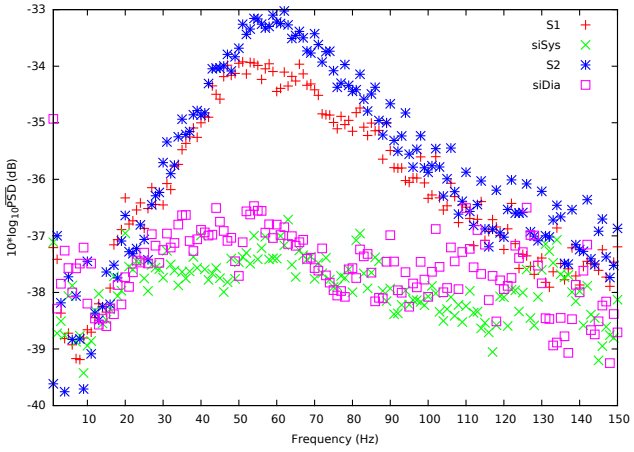


Figure 1: Average power spectral density (PSD) for each state over the frequency range $[0, 150]$ Hz. The $S1$ peak is ≈ 50 Hz and the $S2$ peak is ≈ 60 Hz.

Figure 1. Similar pre-processing methods are also used in [10]. From the filtered signal, the homomorphic envelogram is computed as in [3]. In our previous work [7], we experimented several different envelograms and confirmed that the homomorphic envelogram is suitable for pediatric heart sound signals.

3.3 Performance metrics

The performance of the HSMM and HMM was measured as the model's capacity to recreate the state sequence annotated by the cardiacpulmonologists. We first compute the positive predictability per sample (P^+_{sample}). A sample at time t is a true positive when the predicted state sample and the annotated state sample are the same ($s_t^{model} = s_t^{expert}$), otherwise it is a false positive. Another metric adopted is the positive predictability per state (P^+_{state}). A classification is a true positive when the model's state is equal to the closest expert's annotated state. Otherwise it is considered a false positive.

4 Results

We conducted experiments both with HMM and HSMM. The HMM is not as capable as the HSMM in detecting the right sequence and duration of states, as it can be seen in Figure 2, where the HMM average positive predictability per sample P^+_{sample} and per state P^+_{state} is considerably lower than the P^+_{sample} and P^+_{state} of any HSMM that we tested. As it is shown in Figure 2, the HSMM using the Poisson distribution outperformed significantly the ones based on Gaussian, Gamma distributions and the non-parametric probability mass function for P^+_{sample} and P^+_{state}. Furthermore, we can see that the non-parametric probability mass function, while it starts with weak performance, it improves significantly as the size of the training set increases.

5 Conclusion

In this paper, we used HSMMs to decode the "true" state sequence of events in a PCG signal. We observed, that the HSMM always outperforms the HMM. We use an ensemble of distributions to approximate the sojourn time distribution. Our experiments using a subject dependent approach, showed that the Poisson clearly outperformed the Gaussian and Gamma distribution and the non-parametric probability mass function. We concluded that using information concerning the sojourn time distribution in each state is a compulsory step when modeling heart sound signals.

6 Acknowledgment

This article is a result and funded by the project NanoSTIMA, NORTE-01-0145-FEDER-000016, supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). It is also a result of internal project SmartHeart in scope of project UID/EEA/50008/2013. This work was also funded by the FCT grant SFRH/BPD/118714/2016. T. Mantadelis is funded by SMILeS (PL02024) project.

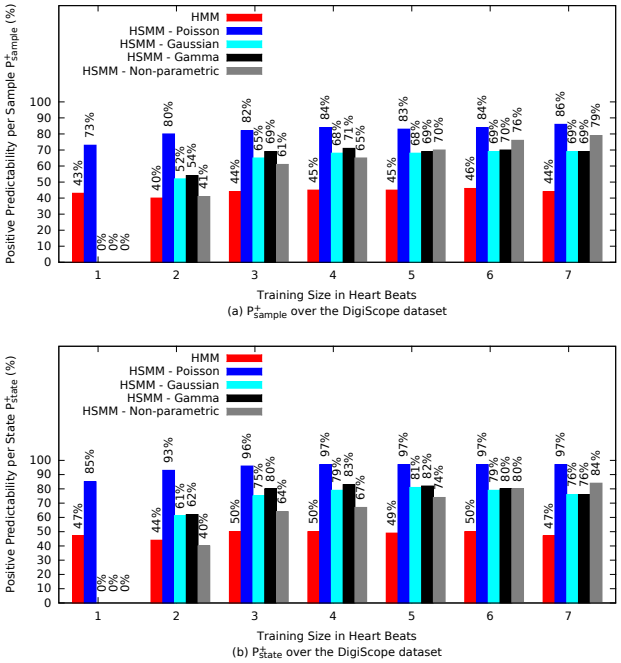


Figure 2: Subject dependent results. Average positive predictability (a) per sample P^+_{sample} and (b) per state P^+_{state} for the tested HMM, HSMM models over the DigiScope dataset.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[2] G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61 (3):268–278, 1973.

[3] D. Gill, N. Gavrieli, and N. Intrator. Detection and identification of heart sounds using homomorphic envelogram and self-organizing probabilistic model. In *Computers in Cardiology*, pages 957–960, 2005. doi: 10.1109/CIC.2005.1588267.

[4] Cota Navin Gupta, Ramaswamy Palaniappan, Sundaram Swaminathan, and Shankar M. Krishnan. Neural network classification of homomorphic segmented heart sounds. *Appl. Soft Comput.*, 7(1): 286–297, 2007. ISSN 1568-4946. doi: 10.1016/j.asoc.2005.06.006.

[5] John Edward. Hall and Arthur C. Guyton. *Textbook of medical physiology*. Saunders/Elsevier, Philadelphia, Pa., 12th edition, 2011.

[6] H. Liang, S. Lukkarinen, and I. Hartimo. Heart sound segmentation algorithm based on heart sound envelogram. In *Computers in Cardiology*, pages 105–108, 1997. doi: 10.1109/CIC.1997.647841.

[7] J. Oliveira, T. Mantadelis, and M. Tavares Coimbra. Why should you model time when you use markov models for analysing heart sounds. In *IEEE EMBC*, 2016.

[8] D. Pereira, F. Hedayioglu, R. Correia, T. Silva, I. Dutra, F. Almeida, S.S. Mattos, and M. Coimbra. DigiScope - Unobtrusive collection and annotating of auscultations in real hospital environments. In *IEEE EMBC*, pages 1193–1196, 2011. doi: 10.1109/IEMBS.2011.6090280.

[9] S.E. Schmidt, E. Toft, C. Holst-Hansen, C. Graff, and J.J. Struijk. Segmentation of heart sound recordings from an electronic stethoscope by a duration dependent hidden-markov model. In *Computers in Cardiology*, pages 345–348, 2008. doi: 10.1109/CIC.2008.4749049.

[10] David B. Springer, Lionel Tarassenko, and Gari D. Clifford. Logistic regression-hsmm-based heart sound segmentation. *IEEE Transactions on Biomedical Engineering*, 63(4):822–832, 2016.

Paper:#26

An Expression-specific Deep Neural Network for Emotion Recognition

By: Ferreira, P. Marques, F. Cardoso, J. Rebelo, A.

An Expression-specific Deep Neural Network for Emotion Recognition

Pedro M. Ferreira [1,2]
pmmf@inesctec.pt
Filipe Marques [2]
fmmarques16@gmail.com
Jaime S. Cardoso [1,2]
jaime.cardoso@inesctec.pt
Ana Rebelo [1,3]
arebelo@inesctec.pt

[1] INESC TEC
Porto, Portugal
[2] Universidade do Porto
Porto, Portugal
[3] Universidade Portucalense
Porto, Portugal

Abstract

Facial expression recognition (FER) is currently one of the most active research topics due to its wide range of applications in the human-computer interaction field. An important part of the recent success of automatic FER was achieved thanks to the emergence of deep learning approaches. However, training deep networks for FER is still a very challenging task, since most of the available FER datasets are relatively small. In this regard, we propose a novel deep neural network architecture along with a well-designed loss function that explicitly models both informative local facial regions and expression recognition. The result is a model that is able to jointly learn facial relevance maps and expression-specific features for a proper recognition. Experimental results demonstrate the effectiveness of the proposed model in both lab-controlled and wild environments.

1 Introduction

Automatic facial expression recognition (FER) has been one of the key problems in the human-computer interaction field, with growing application areas including neuromarketing, crowd analytics, biometrics or clinical monitoring [1]. Expression recognition is a task that human beings perform daily and effortlessly, but it is not yet easily performed by computers. Although recent methods, particularly those using deep learning, have demonstrated remarkable performances in highly controlled environments, the automatic FER in real-world scenarios is still a very challenging task [1]. In addition, the performance of deep models is still below of its full potential as training high capacity models in small datasets, such as the ones available in the FER field, usually result in overfitting.

To work around the problem of training high-capacity classifiers on small datasets, previous FER works have mainly resorted to (i) *transfer learning* [3], where a CNN is typically pre-trained in some domain-related dataset before being fine-tuned to the target dataset; and (ii) *classifier ensembles* [2], in which an ensemble of CNNs is created in order to combine their decisions and, hence, reduce the model's variance. However, their benefits are tightly coupled with the source-target domain similarity.

Inspired by the strong support from physiology and psychology that FEs are the result of the motions of facial muscles [1], a novel end-to-end deep neural network along with a well-designed loss function for FER are proposed. The loss function is defined in a such manner to regularize the entire learning process, so that the proposed model is able to automatically learn expression-specific features and, hence, improve the generalization capability of the model.

2 Proposed Method

In this paper, we propose a novel deep neural network architecture along with a well-designed loss function that explicitly models both informative local facial regions and expression recognition. The underlying idea is to explicitly drive the model towards the most relevant facial areas for the expression recognition, such as the facial components (i.e., eyes, eyebrows, nose, mouth) and expression wrinkles. To accomplish this purpose, the proposed neural network is composed by three main components, namely (i) the *facial-parts component*, (ii) the *representation component*, and (iii) the *classification component* (see Figure 1). The purpose of the *facial-parts component* is to learn an encoding-decoding function $E(\mathrm{x})$ that maps from an input image x to a relevance map $\hat{\mathrm{x}}$ representing the probability of each pixel being relevant for recognition. The loss function is defined in a such manner that enforces sparsity and spatial

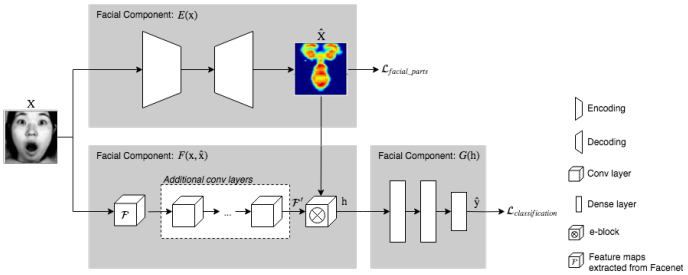


Figure 1: Architecture of the proposed model.

contiguity on the activations of $\hat{\mathrm{x}}$. This definition is supported by the physiological fact that just small and disjoint facial regions are relevant for recognition [1]. The *representation component* aims to learn an embedding function $F(\mathrm{x})$ that maps from an input image x and its relevance map $\hat{\mathrm{x}}$ to an hidden representation h. The relevance map $\hat{\mathrm{x}}$ that is being learned in the *facial-parts component* is then used to filter the learned representations h, enforcing them to only respond strongly to the most relevant facial parts as possible. The result is a model that produces highly discriminative representations for FER. The *classification component* is then trained on these highly discriminative representations.

2.1 Architecture

2.1.1 Facial-parts component

The *facial-parts component* consists of a convolutional path (or encoding) followed by a deconvolutional path (or decoding), in a such way that it is possible to learn a mapping between an input image x to a relevance map $\hat{\mathrm{x}}$, with the same resolution of the input. The encoding comprises several sequences of two consecutive 3×3 convolutional layers, with rectified linear units (ReLUs) as non-linearities, followed by a 2×2 max-pooling operation for downsampling. Every step in the decoding comprises a 2×2 transpose convolution and two 3×3 convolutions, each one followed by a ReLU The transpose convolution is applied for up-sampling and densify the incoming features maps. At the final layer a 3×3 convolution with a linear activation function is used to map the activations into a probability relevance map.

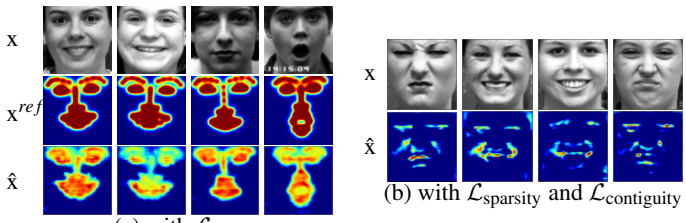
2.1.2 Representation component

The purpose of the *representation component* is to extract highly discriminative features for FER. Therefore, the image is first analyzed by a convolutional network (initialized by the first 10 layers of Facenet [1] and finetuned), generating a set of feature maps $\mathcal{F}$ that is input to a set of additional convolutional layers. Since $\mathcal{F}$ came from a pre-trained network of a similar but different domain (i.e., face recognition), the additional convolutions assure the computation of more complex and high level features for FER. The resulting feature maps, $\mathcal{F}'$, are then used as input in a novel building block of the network, the so-called expression block (e-block), in order to increase the discriminative ability of the learned features. The e-block performs an elementwise multiplication between the learned features $\mathcal{F}'$ and the relevance map $\hat{\mathrm{x}}$, to enforce the output activations h to just respond strongly to the most relevant facial parts.

2.1.3 Classification component

The *classification component* simply consists of a sequence of fully connected layers (or dense layers). The last layer of the CNN is a softmax output layer, which contains the output probabilities for each class label.

[1] Facenet: A unified embedding for face recognition and clustering. https://arxiv.org/abs/1503.03832



(a) with $\mathcal{L}_{\text{kpts}}$ (b) with $\mathcal{L}_{\text{sparsity}}$ and $\mathcal{L}_{\text{contiguity}}$

Figure 2: The effect of each facial-parts loss term on the relevance maps.

2.2 Learning

Inference in the proposed model is given by $\hat{x} = E(x)$ and $\hat{y} = G(h)$ where $\hat{x}$ is the relevance map of the facial parts, $\hat{y}$ is the task-specific prediction and $h = F(x, \hat{x})$. Therefore, the goal of training is to minimize the following loss function with respect to parameters $\Theta = \{\theta_E, \theta_F, \theta_G\}$:

$$\mathcal{L} = \mathcal{L}_{\text{classification}} + \lambda\ \mathcal{L}_{\text{facial\_parts}}, \tag{1}$$

where $\lambda \geq 0$ is the weight that controls the interaction of the loss terms. The classification loss, $\mathcal{L}_{classification}$, trains the model to predict the categorical emotions $\hat{\mathbf{y}}$ given the ground-truth $\mathbf{y}$ and corresponds to the categorical cross-entropy.

The purpose of the facial-parts loss, $\mathcal{L}_{facial\_parts}$, is to enforce the relevance map $\hat{x}$ to encode the relative importance of each pixel to the facial expression classification. The underlying assumption is that the relevance map $\hat{x}$ should be sparse and spatially localized. It means that $\hat{x}$ should take high values just in the neighborhood of important facial components (e.g., eyes, eyebrows, nose, mouth and expression winkles). To accomplish this purpose, $\mathcal{L}_{facial\_parts}$ is defined as a balance between map supervision and contiguity and sparsity impositions:

$$\mathcal{L}_{\text{facial\_parts}} = \sum_{i=1}^{N} \mathcal{L}_{\text{kpts}}(\mathbf{x}_i^{ref}, \hat{\mathbf{x}}_i) + \beta \sum_{i=1}^{N} \mathcal{L}_{\text{sparsity}}(\hat{\mathbf{x}}_i) + \gamma \sum_{i=1}^{N} \mathcal{L}_{\text{contiguity}}(\hat{\mathbf{x}}_i), \tag{2}$$

where $\beta, \gamma \geq 0$ are the weights that controls the relative importance of each loss term.

The key-points loss, $\mathcal{L}_{\text{kpts}}$, encourages the relevance map $\hat{x}$ to take high values in the neighborhood of the most important facial components (i.e., eyes, eyebrows, nose and mouth). To accomplish this purpose, a target (or reference) relevance map $\mathbf{x}^{ref}$ is created, based on the ground-truth coordinates of the facial landmarks. That is, for each training image, each key-point is represented by a Gaussian (with mean at the key-point coordinates and predefined standard deviation), and the target relevance map $\mathbf{x}^{ref}$ is formed by the mixture of the Gaussians of each facial landmark. $\mathcal{L}_{\text{kpts}}$ is then defined to minimize the mean squared error between the target and the predicted relevance maps, such that:

$$\mathcal{L}_{\text{kpts}} = \frac{1}{m \times n} \sum_{i,j} (\mathbf{x}_{i,j}^{ref} - \hat{\mathbf{x}}_{i,j})^2, \tag{3}$$

where m, n denote the resolution of the relevance map $\hat{\mathbf{x}}$.

The last two loss terms impose sparsity and contiguity constrains on the activations of $\hat{\mathbf{x}}$, so that the predicted relevance maps may have the potential to capture other important facial clues, such as the expression wrinkles or dimples, that are not located in the neighbourhood of the facial landmarks. The intuition is that just small and disjoint facial regions are relevant for the recognition task. The sparsity term is then defined by:

$$\mathcal{L}_{\text{sparsity}}(\hat{\mathbf{x}}) = \frac{1}{m \times n} \sum_{i,j} |\hat{\mathbf{x}}_{i,j}|, \tag{4}$$

The spatial contiguity term $\mathcal{L}_{\text{contiguity}}$ encourages the activations of $\hat{\mathbf{x}}$ to be smooth and spatially localized, by minimizing the local spatial transitions of the relevance map $\hat{\mathbf{x}}$:

$$\mathcal{L}_{\text{contiguity}}(\hat{\mathbf{x}}) = \frac{1}{m \times n} \sum_{i,j} |\hat{\mathbf{x}}_{i+1,j} - \hat{\mathbf{x}}_{i,j}| + |\hat{\mathbf{x}}_{i,j+1} - \hat{\mathbf{x}}_{i,j}| \tag{5}$$

By combining all the three loss terms, the predicted relevance maps $\hat{x}$ will encode local appearance information around facial landmarks with the freedom to capture additional sparse and contiguity facial features, such as expression wrinkles and dimples.

Table 1: Results on CK+ and SFEW datasets: (first block) baseline methods; and (second block) variations of our method.

Method	CK+		SFEW	
	Acc (%)	Loss	Acc (%)	Loss
CNN from Scratch	88.60	0.58	36.01	1.88
Facenet finetuned	93.75	0.28	46.02	**1.57**
VGG16 finetuned	91.67	0.42	-	-
Proposed with $\mathcal{F}$ from Scratch	91.11	0.51	-	-
Proposed with $\mathcal{F}$ from Facenet	**94.21**	**0.20**	**47.26**	1.79
Proposed with $\mathcal{F}$ from Facenet + refinement	93.85	0.23	-	-

2.3 Iterative Refinement

The predicted relevance maps $\hat{\mathbf{x}}$ have a crucial role in the final classification since they are merged with the computed features $\mathcal{F}$. In this regard, an iterative refinement strategy, for improving the estimation of $\hat{\mathbf{x}}$, was implemented. That is, the feature space returned by the e-block, h, will be the input for a new stage. For stage >= 1, the feature space from the first encoding $\mathcal{F}$ is supplied to each merge operation, allowing the classifier to freely combine contextual information by picking the most predictive features. A new and more refined map is generated in each stage.

3 Experimental Results

The experimental evaluation of the proposed deep neural network was performed using two public available databases for FER: (i) the CK+ , a lab-controlled dataset, and (ii) the SFEW 2.0, a dataset with spontaneous expressions under wild (non-controlled) scenarios. The hyperparameters off all the models, including the baselines, were optimized by means of grid search and cross-validation on the training set. During the training stage, several traditional regularization techniques were also applied (i.e., dropout and data augmentation).

Experiments on CK+ and SFEW databases are presented in Table 1, in which a comparison between the proposed model and baseline methods is performed. The results are presented in terms of average accuracy (Acc) and loss. We consider the fine-tuned VGG-16 and Facenet as our baselines. To further demonstrate the effectiveness of the proposed method, we also include the results of a CNN trained from scratch with the architecture of the *representation component* of our model. For both databases, our method significantly outperforms all others. Regarding the variations of the proposed model, it is possible to observe that is better to initialize the *representation component* with feature maps extracted from Facenet rather than from the scratch. The iterative refinement strategy does not promote any improvements in the overall classification.

Figure 2 depicts the effect of each facial-parts loss term in the learned relevance maps $\hat{x}$. As expected, the activations of $\hat{x}$ are strong in the neighborhood of important facial components. This demonstrates that the relevance maps are suitable to enforce the model to learn highly discriminative representations for FER. Using $\mathcal{L}_{\text{kpts}}$, the resulting relevance maps $\hat{x}$ "just" encode the local appearance information around the facial landmarks (see Figure 2 (a)). Interestingly, by imposing both sparsity and contiguity impositions, the resulting relevance maps are also able to capture expressions wrinkles and dimples (see Figure 2 (b)).

4 Conclusions

In this paper, we propose a novel end-to-end deep neural network architecture along with a well-designed loss function that jointly learn the most relevant facial parts along with the expression recognition. The result is a model that is able to learn expression-specific features. Experimental results on two well-known facial expression databases CK+, and SFEW demonstrate the potential of the proposed model in both lab-controlled and wild scenarios.

Acknowledgements This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia as part of project «UID/EEA/50014/2013» and within PhD grant number «SFRH/BD/102177/2014».

References

[1] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero. Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1548–1568, 2016.

[2] Bo-Kyeong et al. Kim. Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *Journal on Multimodal User Interfaces*, 10(2):173–189, Jun 2016.

[3] Zhiding Yu and Cha Zhang. Image based static facial expression recognition with multiple deep network learning. ICMI '15, pages 435–442. ACM, 2015.

Paper:#29

Segmentation and Classification of Skin Lesions Based on Texture and YIQ Color Space Features

By: Ribeiro, A. Almeida, P. Almeida, S. Vasconcelos, V. Lopes, F.

Segmentation and Classification of Skin Lesions Based on Texture and YIQ Color Space Features

Patrícia Sousa[1,*]
Alfredo Ribeiro[1,*]
Susana Almeida[1,*]
Verónica Vasconcelos[1,2]
Fernando Lopes[1,3]

[1]Coimbra Institute of Engineering, Polytechnic Institute of Coimbra
[2]INESC TEC
[3]Telecommunications Institute - Coimbra

Abstract

In this paper, we propose a successful approach for automatic segmentation and classification of skin lesions in two classes: melanoma and non-melanoma. Initially, skin images obtained in a clinical environment are pre-processed to remove unwanted hair and reduce noise. A region growing segmentation technique using automatic initialization of seed points is then applied, in order to isolate the lesion areas for further processing. Subsequently, the extracted lesion areas are represented by a set of color and texture features. Using a Support Vector Machine (SVM) classifier, the features extracted from each segmented lesion are organized in a feature vector that is further used to discriminate the lesions between melanoma and non-melanoma. The classifier performance was evaluated using a stratified holdout approach to measure its sensitivity, specificity and accuracy. The best results for accuracy were obtained with the Gaussian kernel and are in the order of 85%. The results are promising compared with similar works [1].

1 Introduction

Melanoma is the more serious type of skin cancer. According to the World Health Organization (WHO), around 132,000 cases of malignant melanoma and more than 2 million of other skin cancers occur each year around the world [2][1]. Melanoma lesions may present a benign form designated as *nevus*, or a malignant form, commonly referred to as *melanoma*. Melanoma is frequently found in the form of a large, pigmented, irregular contour and multicolour skin lesion. Despite the sharp mortality associated with melanoma, early diagnosis is extremely important since 90% of the cases have cure if diagnosed at an early stage. As a result, many efforts have been dedicated to the early diagnosis and many studies focus on developing automated techniques to assist the diagnosis [1] [3] [4]. In this paper we propose a method that combines a set of techniques that allow to classify skin lesions in melanoma and non-melanoma. The method is based on a feature vector using texture, RGB and YIQ color-space characteristics.

2 Dataset

The images of the skin lesions were selected from 58 patients of Pedro Hispano Hospital from Oporto. In the dataset there are 25 images corresponding to melanoma and the remaining to non-melanoma lesions. The both classes exhibit large intra-class variations. The melanoma lesions have irregular borders, are asymmetric, multicolour and with diameter higher than 6 mm. Since there was little control over the image acquisition and camera calibration, the images have different contrast, size, and illumination. In Figure 1 we illustrate two example images from the dataset, one of a melanoma and one of a non-melanoma.

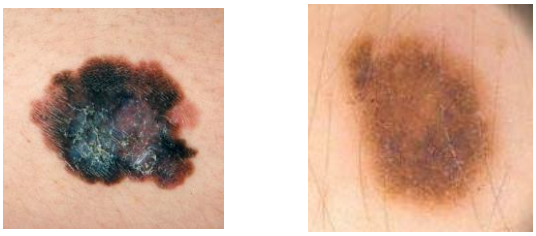


Figure 1: Examples of lesions in the dataset. (Left) Melanoma. (Right) Non-melanoma.

3 Pre-processing and Segmentation of Images

For the feature extraction and classification phases, the pigmented lesions in the original images must be isolated from healthy skin through segmentation. For this purpose, techniques such as region growing, active contours or detection of discontinuities may be used.

Before the segmentation process can be successfully applied, a pre-processing phase is required with the main objective of improving the image quality by removing unwanted artefacts, compensate for large variations in intensity and to improve the contrast.

As a first pre-processing step, a smoothing Gaussian filter was used in order to reduce noise and remove hair from the images. From the RGB filtered image a binary image was created that was further used for a region growing segmentation technique. The results of the segmentation were improved by the use of morphological operations. A close operation allowed to add isolated inner pixels to the lesion foreground while an open operation allowed to remove isolated pixels from the background. A Canny filter was used to detect the lesion contour and create the final region of interest (ROI) image area. The results of the pre-processing and segmentation phase are illustrated in Figure 2, where the superimposed detected contour on the original image (left) and the isolated skin lesion image (right), are shown.

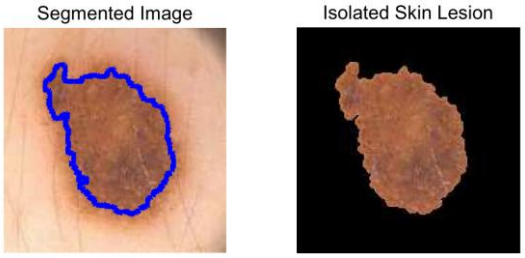


Figure 2: Segmented image and isolated skin lesion image.

4 Feature Extraction

The segmentation of the pigmented skin lesions allows to define the ROI in the original images. To proceed to their classification into two classes (melanoma, non-melanoma) the ROI is described in a quantitative manner through a set of extracted characteristics or features.

In this work we use two sets of features. The first set is extracted from the first moments of the color information in the RGB and YIQ color spaces. The second set is extracted from texture descriptors based on the Gray Level Co-occurrence Matrix (GLCM) [5]. Color and texture are very important characteristics that can be used to convey differences between images of malign and non-malign skin lesions.

In terms of the color features we use both the RGB and YIQ color spaces [6]. The RGB space is based in the human perception of colors and uses three additive primary color components. It is the conventional color space for acquisition, representation and display of images in electronic systems. The YIQ space was defined by the National Television Systems Committee (NTSC) and consists in the recoding of non-linear RGB for television transmission efficiency. In this system, the image data consists of the *Luminance* (Y), *In-phase* (I) and *Quadrature* (Q) components. Y represents the grayscale image information, while I and Q represent the chrominance or image color information. By selecting the YIQ color space we can have I representing the orange-blue color range and Q representing the

[*] Master's Degree in Biomedical Instrumentation Student - ISEC

purple-green color range (Figure 3). Both these color ranges are very characteristic of skin lesion images.

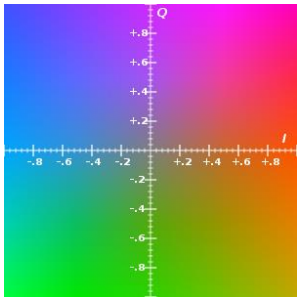


Figure 3: YIQ color space at Y=0.5.

The color features are calculated for each color channel and associated with color moments. The first four colour moments were computed: mean, standard deviation, symmetry, and variance. Considering 4 features per color channel and 2 colour spaces with 3 components each, we obtain a total of 24 color features.

In what refers to the texture features they are extracted from statistical texture descriptors based on the computation of gray level co-occurrence matrices. Four spatial directions are considered for matrix calculation: 0°, 45°, 90° and 135°. For each matrix a set of five characteristics was calculated: contrast, correlation, energy, and homogeneity [5]. Five features were created averaging the similar features for each independent orientation.

5 Classification

The texture, RGB and YIQ color spaces' features extracted from each segmented lesion are organized in a feature vector and used to discriminate the lesions in melanoma or non-melanoma using a Support Vector Machine (SVM) classifier. This classifier is based on the principle of structural risk minimization. It looks for an optimal separating hyperplane that maximizes the margin between the two classes in the training data [7]. When the data are not linearly separable in the input space, feature vectors are mapped to a higher dimensional space by using kernel functions. In this work the Gaussian kernel was used, defined by $K(x,y) = exp(-\|x-y\|^2/(2\sigma^2))$ where σ is the Gaussian width.

To assess the classifier performance, the dataset was divided into train and test, using a stratified holdout. Considering a melanoma lesion a positive sample, evaluation was performed in the test set through: sensitivity, that is the ratio between the samples correctly classified as positive (true positive) and the total number of positive samples (true positive plus false negative); specificity, that corresponds to the ratio between the number of samples correctly classified as negative (true negative) and the total number of negative samples (true negative plus false positive), and the accuracy, a global measure defined as the total number of correctly classified samples (true positive plus false positive) divided by the total number of samples [7].

6 Results

The dataset was randomly divided in train and test sets, in a proportion of 50%. The samples of each set were selected using a holdout strategy with stratification, which assures roughly the same class proportions as in the initial dataset.

During the training of the SVM classifier, a search for optimal parameters was performed in the train set. In the case of linear SVM, parameter *C* was tuned. This parameter corresponds to a penalty over the training errors. For the nonlinear SVM, the parameter σ from the Gaussian kernel was also tuned.

The performance of the classifier was evaluated in the independent test set. The best classification values for linear kernel was obtained with *C* = 1, with a sensitivity value of 93.3%, specificity of 66.6% and accuracy of 81.5%. Considering a Gaussian kernel, the parameters that maximize the performance of the classifier were (C=10; σ=1), allowing for a sensitivity value of 93.3%, specificity of 75.5%, and accuracy of 85.2%. Table 1 summarizes the obtained values.

Table 1. Values of sensitivity, specificity, and accuracy using linear and Gaussian kernels. Values in percentage.

	Kernel	
	Linear	Gaussian
Sensitivity	93.3	93.3
Specificity	66.6	75.5
Accuracy	81.5	85.2

The implemented classifier is sensitive to the presence of melanoma lesions. Both kernels allow for a high sensibility, one of the most important metrics in computer-aided diagnosis systems, since in 93.3% of the skin lesions the presence of melanoma is signalized. However, the number of false positives, e.g. non-melanoma lesions that are classified as melanoma, is high, resulting in low specificity values, especially when linear SVM was used. Taking into account the low number of samples of the dataset and the large intra-class variation, the results show the potential of the extracted features, obtained from the combination of texture and color descriptors, used to characterize the segmented skin lesions.

Conclusion and Future Work

In this work, we proposed, implemented and tested an automated method for the segmentation and classification of melanoma-type skin lesions. The method was developed and tested using an image dataset obtained in a clinical environment.

A pre-processing and segmentation phase used Gaussian filtering, region growing and morphological operators to isolate the lesions and generate the ROIs for further processing. In this process four images were discarded due to unsuccessful segmentation that would negatively influence the results of the SVM. A feature vector was created using 5 texture and 24 RGB and YIQ color spaces' features. An SVM classifier with a Gaussian kernel was very effective in the classification of the melanoma lesions, achieving an accuracy of 85%.

Considering the small dataset and its large intra-class variation, the obtained results allow us to conclude that the proposed method, including the specific use of the YIQ color space, has a strong potential for automated classification of melanoma-type skin lesions and thus to be used in computer-aided diagnosis systems. In the future we aim to improve the global classification, including increasing the specificity, using other validation strategies, such as *leave-one-out*, and investigating in more detail the benefits of the YIQ color space.

References

[1] R. Sumithra, M. Suhil, and D. S. Guru. Segmentation and classification of skin lesions for disease diagnosis. *Procedia Computer Science*, 45: 76-85, 2015.

[2] World Health Organization (2018). *Ultraviolet radiation (UV) – Skin Cancer*, Accessed on: 10th September 2018. [Online]. Available: http://www.who.int/uv/faq/skincancer/en/index1.html.

[3] J. E. McWhirter, and L. Hoffman-Goetz. Visual images for patient skin self-examination and melanoma detection: a systematic review of published studies. *Journal of the American Academy of Dermatology*, 69.1: 47-55, 2013.

[4] Y. Li, and L. Shen. Skin lesion analysis towards melanoma detection using deep learning network, *Sensors*, 18(2): 556, 2018

[5] R. M. Haralick. Statistical and structural approaches to texture. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5): 786-804, 1979.

[6] S. A. Machekposhtia, M. Soltani, K. Raahemifarc, E. Z. Bidakid, and M. Sadeghie. PASI area and erythema scoring using YIQ color space, *J. of Dermatology Research and Skin Care*, 1(1): 8-14, 2017.

[7] V. Vapnik. *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 1995.

[8] C. C. Chang, C. J. Lin. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology*, 2(3), 27, 2011.

[9] Y. Lee, Y. Seo, J. B., Lee, J. G., Kim, S. S., Kim, N., and Kang, S. H. Performance testing of several classifiers for differentiating obstructive lung diseases based on texture analysis at high-resolution computerized tomography (HRCT). *Computer methods and programs in biomedicine*, 93(2): 206-215, 2009.

Paper:#31

Speckle Noise Reduction in Medical Ultrasound RF Raw Images

By: Santo, V. Monteiro, F.

Speckle Noise Reduction in Medical Ultrasound RF Raw Images

Verónica Espírito Santo
a29441@alunos.ipb.pt
Fernando C. Monteiro
monteiro@ipb.pt

School of Technology and Management,
Polytechnic Institute of Bragança
CeDRI, Research Centre in Digitalization and Intelligent Robotics,
Polytechnic Institute of Bragança

Abstract

Ultrasonography is the commonly used imaging modality for the examination of several pathologies due to its non-invasiveness, affordability and easiness of use. However, ultrasound images are degraded by an intrinsic artifact called 'speckle', which is the result of the constructive and destructive coherent summation of ultrasound echoes. This paper aims to generate B-mode images out of radiofrequency (RF) data following standard procedures, a series of steps such as envelope detection, log-compression and scan conversion. The best set of parameters of this pipeline will be selected in order to achieve B-mode images with high quality.

1 Introduction

Ultrasound imaging is one of the most important and cheapest instrument used for diagnostic purpose among the clinicians. However, the images obtained through this type of examination presents a characteristic noise type, known as speckle noise, which makes it difficult to analyze and diagnose [1,2].

Speckle reduction methods can be classified in two categories: image compounding and image filtering [3]. Image compounding is achieved by averaging a series of ultrasound images acquired from different viewpoints. The main drawback is the need of multiple acquisitions. Image filtering techniques include adaptive filters, anisotropic diffusion and wavelets [2].

Recently, new types of filters have been proposed to remove speckle noise from RF data. In [4], the authors used a low pass frequency-shift, followed by a least mean square adaptive filter. Al-Asad [5] proposed a Short Time Fourier transform applied to the envelope of each RF line before reconstruction, followed by its application to the lateral dimension of the 2D image after reconstruction.

In this paper we intend to filtering of the raw RF data to reduce noise and limit the signal to the working bandwidth. This will be done by the application of denoising filters individually to the one-dimensional RF envelopes that will constitute the B-mode image. By filtering RF data in the process of B-mode image construction, we expect to obtain images with less speckle noise.

2 Methods

In order to obtain an ultrasound image in B-mode, an RF signal is received which passes essentially through three signal processing phases, as shown in Fig. 1: IQ Demodulation, Envelope Detection, and Log compression.

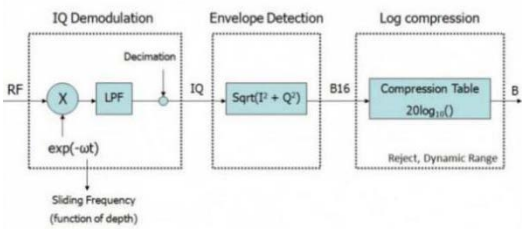


Figure 1: B-mode Signal Processing

IQ Demodulation: is a common and useful technique in RF signal processing. In the demodulation process for digital signals, the waveform is not important because it is already known. The problem comes down to whether the pulse is present or absent, so channel noise has no influence in that direction. However, channel noise may cause certain errors in decisions. The decision on the detection can be facilitated through the passage of the signal through filters that reinforce the useful signal and suppress the noise at the same time. This allows to greatly improve the signal-to-noise ratio by reducing the possibility of error.

Envelope Detection: is one of the simplest analog demodulation techniques in which an electronic circuit that receives a RF signal as input and provides an envelope of the input signal as an output.
Absolute and Hilbert transforms are common methods of envelope detection. It returns the absolute or Hilbert amplitude variation of the amplitude of a time wave.
Logarithmic compression: is the method by which the high amplitudes are compressed at the same time that the signal of the low amplitudes is amplified. This method allows you to view low-amplitude signals on the monitor that would otherwise not be seen.
After obtaining the B-mode image, it goes through a **scan conversion** (Fig. 2). It is a method that helps convert linear B-scan data into geometrically correct images, "fan-shaped" images.

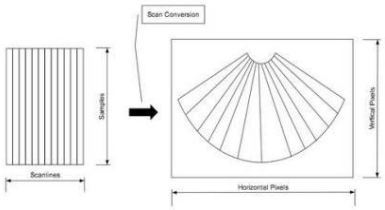


Figure 2: Scan Conversion

The RF signal processing is performed in the steps: IQ Demodulation and Envelope Detection. In these steps, we are allowed to filter the noise present in the RF signal, as shown in Fig. 3.

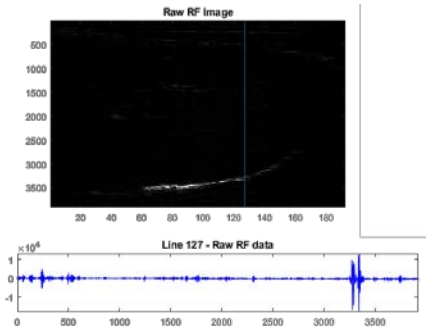


Figure 3: Raw RF Image

In the IQ Demodulation stage, it is possible to apply low pass filters (LPF), that attenuate or reduce the amplitude of the frequencies larger than the cutoff frequency. The amount of attenuation for each frequency varies from filter to filter. And the technique of **downsampling** or **decimation** it is also applied, which allows us to reduce the sampling rate. This is done by simply separating one sample at each N, which can cause signal distortion.
In the envelope detection step, you can apply one of the following options: *Hilbert* (discrete-time analytic signal using Hilbert transform) or *Absolute* (absolute value and complex magnitude).

3 Results

Figure 4 shows the influence of the different filters (*Butterworth, Bessel and, Chebyshev*) and envelope detection (*Hilbert or Absolute*) applied to the raw RF image. By comparing the images obtained and the corresponding signal representation, we can observe that the *Butterworth and Chebyshev* filters are the ones with the best results in noise reduction, along with the *Hilbert Envelope.*
To study the influence of decimation, it was used the combination ***Butterworth filter + Envelope Hilbert***. We considered that this was the combination that shows the best results for noise reduction when compared to the others in the study, as shown in Fig. 5.

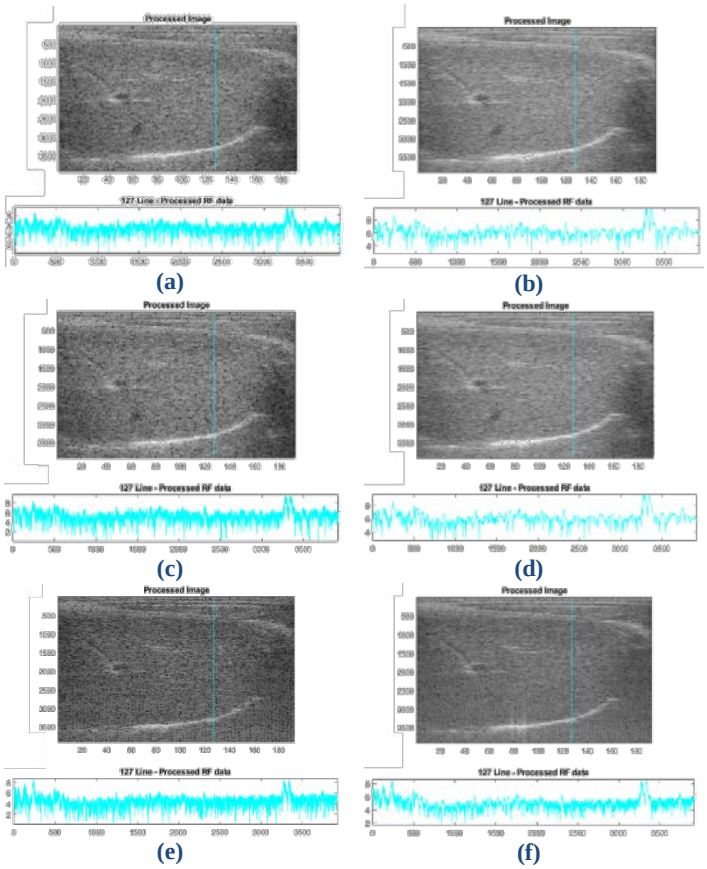


Figure 4: Results with downsampling = 1. **(a)** Butterworth filter, Absolute envelope, log compression **(b)** Butterworth filter, Hilbert envelope, log compression **(c)** Chebyshev filter, Absolute envelope, log compression **(d)** Chebyshev filter, Hilbert envelope, log compression **(c)** Bessel filter, Absolute envelope, log compression (d) Bessel filter, Hilbert envelope, log compression

Figure 5 shows the influence of different downsampling values in B-mode images. In this study, the downsampling assumes the values of 20, 30, 40 and 50.

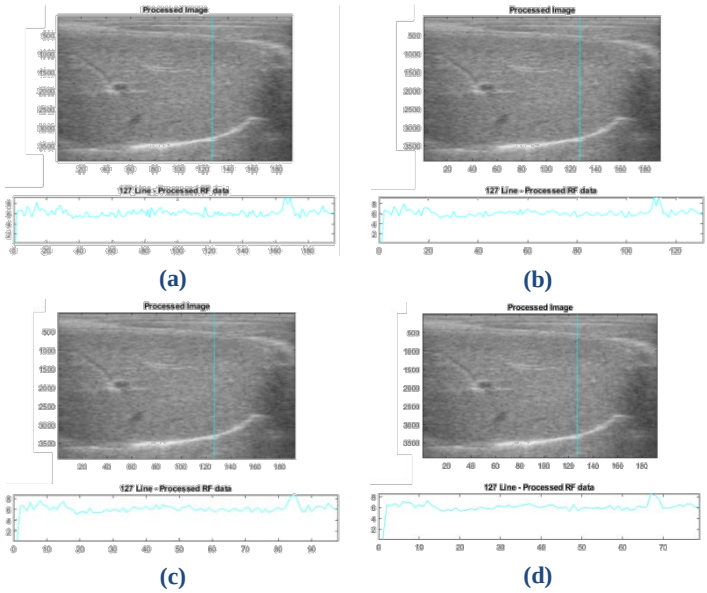


Figure 5: **(a)** Butterworth filter, Hilbert envelope, downsampling = 20, log compression, **(b)** Butterworth filter, Hilbert envelope, downsampling = 30, log compression, **(c)** Butterworth filter, Hilbert envelope, downsampling = 40, log compression, **(d)** Butterworth filter, Hilbert envelope, downsampling = 50, log compression

For downsampling values of 20 and 30, a major improvement in image noise is observed[1], however, for values greater than 30, an undesirable effect happens that causes image distortion, known as *blurring effect*. After testing different filters, downsampling values, and different Envelope detection, it was concluded that the following combination was the one with the best results in the processing of the RF image:

Butterworth filter + Hilbert envelope + downsampling = 30

To justify our choice, we use the image result from this combination as reference to calculate the peak signal-to-noise ratio for the images in Fig.4 and Fig.5. PSNR high means good quality and low means bad quality image. Table 1 shows that the best quality images are obtain with Butterworth filter combine with Hilbert envelope. And also shows the bigger the downsampling value the better quality. However, visually for downsampling value of 40 the image starts to become affected by the undesirable *blurring effect*. That's why we considered downsampling 30 the best option.

Filters	DownSampling	Envelope	PSNR
Butterworth	40	Hilbert	34,5883
	20		32,4544
	1		23,9061
Chebyshev			23,8879
Bessel			17,6883
Butterworth		Absolute	18,1407
Chebyshev			17,7704
Bessel			14,3964

Table 1 Peak signal-to-noise ratio

Median filter was applied to the B-mode image. This filter is one of the image processing techniques used for removing speckle noise. Figure 6 shows the RF and the B-mode processed images with contrast adjustment. By comparing this two images it is possible to affirm that RF signal processing shows to be equally efficient in noise reduction as the B-mode filtering.

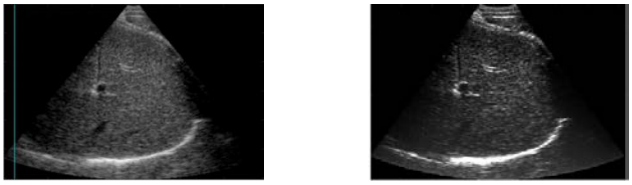


Figure 6: **(a)** Scan convert RF Image, Butterworth Filter, Hilbert Envelope, Downsampling=30, and contrast adjustment, **(b)** Scan convert B-mode Image filtered with Median filter and contrast adjustment.

4 Conclusions

In this paper we proposed and approach to reduce speckle noise in ultrasound images by filtering the raw RF data, before obtain the B-mode image. This was done by the application of denoising filters individually to the 1D RF envelopes that will constitute the B-mode image.

From the results we can conclude that filtering in RF mode, before the conversion to B-mode, reduce the speckle noise in B-mode image. This approach needs to be study in order to reduce even more the speckle noise.

References

[1] S. H. Contreras Ortiz, T. Chiu, and M. D. Fox, "Ultrasound image enhancement: A review," Biomedical Signal Processing and Control, vol. 7, no. 5. pp. 419–428, 2012.

[2] F. C. Monteiro, J. Rufino, and V. Cadavez, "Towards a comprehensive evaluation of ultrasound speckle reduction," in Lecture Notes in Computer Science, 2014, vol. 8814, pp. 141–149.

[3] D. Mittal, V. Kumar, S. C. Saxena, N. Khandelwal, and N. Kalra, "Enhancement of the ultrasound images by modified anisotropic diffusion method," Med. Biol. Eng. Comput., vol. 48, no. 12, pp. 1281–1291, Dec. 2010.

[4] S. Wang, C. Li, M. Ding, and M. Yuchi, "Frequency-shift low-pass filtering and least mean square adaptive filtering for ultrasound imaging," in Progress in Biomedical Optics and Imaging - Proceedings of SPIE, 2016, vol. 9790, p. 97900P.

[5] J. F. Al-Asad, "Despeckling the 2D medical ultrasound image through individual despeckling of the envelopes of its 1D radio frequency echo lines by STFT," Journal Image Graphics, vol. 4, pp. 67–72, 2016.

[1] https://drive.google.com/open?id=1sCwI8WqLu-tJ3XTuyQz4AKQitQQt6hV7

Paper:#32

Deep Learning versus Classical Machine Learning in Landmine Detection from IR images

By: Guerra, I. Silva, J. Bioucas-Dias, J.

Deep Learning versus Classical Machine Learning in Landmine Detection from IR images

Ivo Fernando Fontes Linhas Guerra — Military Academy; and Instituto Superior Técnico, Lisboa, Portugal

José Silvestre Silva — CINAMIL and DCEE, Military Academy, Lisboa, Portugal

José Bioucas-Dias — Instituto Telecomunicações and Instituto Superior Técnico, Lisboa, Portugal

Abstract

This work explores the detection of landmines using multispectral images acquired in military context. Two methods are proposed, one using traditional classifiers and the other using Deep Learning methods, namely a Convolutional Neuronal Network (CNN). A quantitative analysis shows that using traditional classifiers gives overall accuracy (OA) above 97% in indoor and outdoor environments for the detection of land mines up to a given depth tested, whereas the adopted deep learning methods present an increase in these values for larger mines and a decrease for smaller ones. These experimental results shed light into the factors that influence the detection of mines and into the merits and demerits of CNN based classification compared with classical methods.

1 Introduction

The problem of landmine clearance is timely, complex, and demanding due to a multiplicity of factors to consider at the time of detection. Because of an increasing number of war zones and conflicts worldwide, the menace of landmines and unexploded ordnances is becoming a very serious problem that is going to affect the involved countries for years to come [1]. For the correct understanding of the problem, it is necessary to have a general basis of the main actors. Based on the NATO doctrine and the United States Army School of Engineering [2], a mine is an explosive device used to destroy or incapacitate people or vehicles, boats, or aircraft. In the last decade, much research has been done about demining, grouped into five groups/families according to their basic operational characteristics, which include electromagnetic technology, acoustic/ seismic technology, explosive technology constituting mines, and technologies with physical contact. Electromagnetic technology, in which the proposed method belongs, corresponds to methods that use electromagnetism, electromagnetic spectrum, or electromagnetic induction as tools of detection as the basis [3]. Using this technology, Makki [4] describes a method whose purpose is to differentiate a landmine from its neighborhood into a multi-spectrum image using Visible and Near-Infrared Infra-Red (VNIR), Short Wave IR (SWIR) and Thermal IR (TIR) bands. To solve the demining problem, it is proposed a method for detecting landmines in multispectral images with the application of machine learning tools namely classical and CNN-based classifiers.

2 Methods

For the landmine detection problem, two methodologies were developed. A classic one that follows the steps of a pattern recognition problem and another one that uses Deep Learning. Based on a classical methodology, the first step is to obtain the data, then implement feature extraction and feature selection, and finally the classification. In order to complement this research, it is used a CNN-based approach, creating a convolutional neural network (CNN), which is particularly useful and promising in image classification problems.

Regarding the classical methodology, in the step of image acquisition, the images are acquired by imaging equipment of the Military Academy. Then, the alignment of the different images is made, and all the images are resampled to have the same ground sampling distance (GSD). After this step, it is defined regions of interest (ROI) to extract first order features [5], second order features, also known as Spatial Gray Level Dependence Method [6], and higher order features, such as Gray Level Run Length features [7]. The next step is the feature selection process that aims to reduce the size of the data set by selecting the statistically relevant features and discarding the irrelevant and/or redundant ones [4]. There are three techniques of feature selection algorithms: filters (extract features from the data set without regard to classification or any other method of learning as a criterion), wrappers (use classifiers/learning techniques to assess which features are statistically relevant) and embedded (aim to combine the advantages of the two previous methods) [8]. One of the most used algorithms (filter type) is the Relief, that estimates the quality of attributes/features according to how well their values distinguish between instances that are near to each other [9].

The problem of discriminating landmines from the background is a binary one. To solve this kind of problems there are several classification techniques that can be used, such as neural networks, support vector machines (SVM), k-Nearest Neighbors (kNN), decision trees, among others classifiers. It is important to understand that in machine learning there are three major categories of algorithms: unsupervised learning (the samples in the training set are not labeled), reinforced learning (aims to learn the behavior of software agents or robots based on feedback from the environment) and supervised learning (the samples in the training set are labeled) [10]. In this work, we follow a supervised learning approach. Each training pattern is characterized by a vector of features and a binary value: {+1} for mine and {0} for not mine. The quality of the classification algorithm is evaluated according to the sensitivity, specificity, overall accuracy, precision, and the F-score [11] [12].

Concerning the approach using Deep Learning, it is mainly an upgrade of a neural network, and performs learning/classification task directly from images [13].This technology has never been used for the demining problem. Since our approach is supervised, we adopt a CNN-based classifier [14].

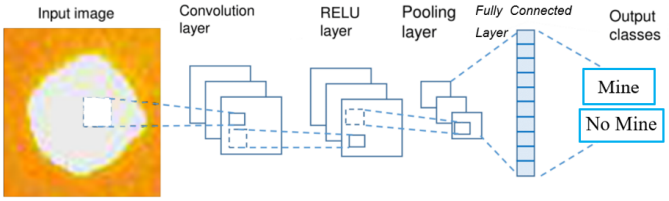


Figure 1: Example of a classification for landmine detection using a CNN-based technique. The input image belongs to TIR band, then it passes to the feature learning step (can be done several times) and finally to the classification part ([14]).

As shown in Fig. 1, a CNN takes an input image of raw pixels, and transforms it via Convolutional Layers, Rectified Linear Unit (RELU) Layers and Pooling Layers (this is called the feature learning step). The output of this transformation feeds into a final Fully Connected Layer which assigns class scores or probabilities, thus classifying the input into the class with the highest probability (called classification step) [14]. The transformation done on the feature learning step are usually repeated many times in order to identify the most individual and particular features possible.

3 Results and discussion

The classification results were produced using a balanced dataset composed by 10262 multispectral AP ROIs extracted from 20 multispectral images and 29984 AT ROIs extracted from 25 multispectral images, both from landmines fields created inside the laboratory (indoor) and 7056 multispectral AP ROIs extracted from 11 multispectral images and 11694 AT ROIs extracted from 10 multispectral images from landmine fields created in the facilities (outdoor) of the Military Academy in Amadora. These results allow to compare the influence of the depths to which the mines are buried, the type of soil in which they are buried, the time they are buried and the type of the landmine, Anti-personal (AP) or Anti-tank (AT). A whole layout was created in order to acquire the necessary multi-spectral images (include: visible band, TIR, and two dub-bands of VNIR) used in the training and in the evaluation of the performance of the classification algorithms.

The next step was the feature selection using the Relief, followed by training the classification algorithms with {26, 66, 132, 198, 264} features and the evaluation of the system performance with the holdout validation method that reserve 15% of the data set as test set. Table 1 shows the best

overall accuracy (OA) and number of features for each classifier, in distinct environments (indoor/ outdoor) and according to the different types of mines (AP and AT). Given the multiplicity of classifiers and machine learning methods used, the OA was the only performance analysis used.

Table 1: Best overall accuracy results and number of features for different classifiers in two distinct environments (indoor / outdoor) and with two types of landmines (AP and AT).

Classification Method	Overall Accuracy [%] / (number of features)			
	AP Indoor	AT Indoor	AP Outdoor	AT Outdoor
Decision Tree	87.0 (264)	94.4 (198)	86.7 (66)	93.9 (198)
Cubic SVM	96.4 (264)	99.0 (198/264)	97.0 (132)	98.7 (264)
Gaussian SVM	97.6 (264)	98.4 (264)	97.5 (132)	97.2 (26)
Fine KNN	94.1 (264)	98.4 (264)	95.1 (198)	97.9 (66/132)
Medium KNN	92.4 (264)	97.9 (264)	93.3 (264)	96.3 (132)
Ensemble Bagged Tree	96.4 (264)	99.1 (198/264)	95.4 (132)	98.4 (132)
Neural Network	90.4 (264)	97.9 (198)	89.6 (66)	96.8 (198)

Once that all the mines present in the data set were buried between 0mm (surface partially in view) and 5mm for AP and between 0mm and 60 mm for AT, the results from Table 1 show that the detection of AT landmines is the one that obtains better results, achieving 99% for Cubic SVM. Compared with the Radial basis function kernel used in the Gaussian SVM, which is frequently used when there is no prior knowledge about the data, we obtained a decrease of 0.6%. The Ensemble Bagged Tree classifier also obtained the best results, with the Decision trees being the worst. This may be since the ensemble uses a technique which combines several decision trees to produce better predictive performance than utilizing a single decision tree. Basically, the main principle behind the ensemble model is that a group of weak learners come together to form a strong learner. The detection of AP landmines is also quite promising, highlighting the Gaussian SVM and ensemble, but achieving slightly lower results. In general, the superior performance of SVM methods confirms the good results these methods have on binary problems It is also verified that many of the best results of OA were obtained with a smaller number of features, which leads us to believe that this reduction of dimensionality makes the method faster and more efficient.

Table 2 presents the results of the Deep Learning CNN network. The network configuration built for the experience, in the feature learning step is composed by: one input layer (size of ROI / dimensions) and three sets of: one convolution layer (filter size, number of filters), one batch normalization layer (used to normalize activations and propagation in the network), one ReLU layer and one pooling layer (down-sampling operation, not used in the third set). In the classification step we have: one fully Connected Layer (this layer is responsible for connecting all the neurons responsible for the features to classify the image), one Soft max layer (normalizes the output) and one classification layer. If we fix the first argument of the convolution layer (filter size) with a value of 3×3 pixels, the 2nd argument, the number of filters, refers to the number of neurons connected to the same input region and thus determines the number of feature maps, which can be varied several times in order to perform a comparative study of OA as a function of the feature map.

Table 2: Best overall accuracy results and number of filters for different environments (indoor/ outdoor) according to the different types of mines (AP and AT). The "n/computed" means that the time required (due to the iterations number) for the training and test this network is too high (more than 7h for training).

Deep Learning (CNN)				
1st layer	8 filters	16 filters	64 filters	256 filters
2nd layer	16 filters	32 filters	128 filters	512 filters
3rd layer	32 filters	64 filters	256 filters	1024filters
	Overall Accuracy			
Indoor AP	82.4	82.7	84.7	86.1
Indoor AT	95.5	97.8	96.7	n/computed
Outdoor AP	79.6	82.0	83.4	82.0
Outdoor AT	99.0	99.1	99.1	n/computed

It is seen that the best results are also related to the detection of AT landmines and that the ideal number of filters / features to be implemented is 64/128/256 respectively for each of the 3 different convolutional layers implemented. There is a 12-20% OA difference between detection of AP mines and AT mines, justified by the fact that AT landmines are larger than AP, allowing to extract more textural information. Regarding the processing time, it was found that the AT input ROIs had the size of 80×80 pixels, while for AP landmines the size of 10×10 pixels making the processing time approximately 5 times lower for ROIs of smaller size. This is because as we define 3×3 pixels the size of the filters for both input images, it is easy to understand that it is more time consuming to compute a filter over an entire 80×80 pixels AT image than an image of 10×10 pixels AP image.

4 Conclusions

Landmines detection is a complex problem that the modern armies and recent technology has not yet solved. Coupled with constant civilian deaths in countries that are, or have already been in conflict, this study analyses the efficiency of landmine detection using multi-spectral image processing. It was shown that for certain depths the values of the OA were in most cases above 97% for the traditional classifiers. The approach of varying the number of filters in the convolution layer of a CNN has the advantage of realizing which feature map is indicated to each specific problem. However, a study of the variation of the number of layers or sets of layers will also have to be performed. In this case, it is verified that a high number of filters for AT makes the network ineffective in terms of time and performance given the large number of feature map that it would have to generate. We thus verified that there should be a balancing of the number of filters for each specific problem. The OA difference between AP and AT landmines shows that this CNN configuration is optimal to 80×80 pixels AT image but not to 10×10 pixels AP image. It may be advisable to obtain larger size ROIs (increasing resolution/decreasing camera distance to ground) or decreasing filter size. However, given the complexity of the problem, it is still early to generalize, since the data set obtained comes from controlled experiences. Thus, there is a need to increase the data set, with more and different landmines fields. It is also demonstrated that there is a promising possibility of developing a deep learning system that would bring the advantages of these types of systems applied to landmine detection.

References

[1] I. Makki, R. Younes, C. Francis e M. Zucchetti, "A survey of landmine detection using hyperspectral imaging," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 124, pp. 40-53, 2017.

[2] Deportment of US Army, "Explosive Hazard Operations," US Army Enginneer School, EUA, pp. 2.1-2.11, 2007.

[3] V. Krylov, "Detection of buried land mines using scattering of Rayleigh waves," em *27th International Conference onNoise and Vibration Engineering (ISMA 2016)*, Leuven, Belgium, 2016.

[4] I. Makki, R. Younes, C. Francis e M. Zucchetti, "Mathematical Methods for Hyperspectral Imaging in Landmine Detection," em *Transactions of the American Nuclear Society*, vol. 112, San Antonio, Texas, 2015.

[5] W. Gonzalez e R. Woods, "Digital Image Processing," Prentice Hall, New Jersey, 2008.

[6] M. S. Priya e G. M. Nawaz, "Matlab Based Feature Extration and Clustering Images using K-Nearest Neighbour Algorithm," *iJact,* vol. 2, pp. 1121-1126, 2016.

[7] M. M. Galloway, "Texture analysis using gray level run lenghts," Computer graphics and image processing, Maryland, EUA, pp. 172-179, 1975.

[8] Z. M. Hira e D. F. Gillies, " A review of feature selection and feature extraction methods applied on microarray data," Advances in bioinformatics, 2015.

[9] N. Morono e A. Betanzos, "Filter Methods for Feature Selection – A Comparative Study," em *Intelligent Data Enginneering and Automated Learning - IDEAL, 8th International Conference*, Birmingham, UK, pp. 178-187, 2017.

[10] Y. Jin e B. Sendhoff, "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," *IEEE Transactions os Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol. 38, pp. 397-415, 2008.

[11] N. Macari, "Analysis of a machine learning algorithm and corpus as a tool for managing the ambiguity problem of search engines," Master of Science, Fakultat Informatik, Technische Universitat Dresden, 2010.

[12] A. R. Webb e K. D. Cospsey, "Statistical pattern recognition," Chichester: John Wiley & Sons, 2011.

[13] H. Greenspan, B. Ginneken e R. Summers, "Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique," *IEEE Transactions on Medical Imaging,* vol. 35, nº 5, pp. 1153-1159, Maio 2016.

[14] J. Ker, L. Wang, J. Rao e T. Lim, "Deep Learning Applications in Medical Image Analysis," *Special Section on fodt Computing Techniques for image analysis in the medical industry current trends, challenges and solutions,* vol. 6, pp. 9375-9389, 2018.

Paper:#33

Pyramid Spatial Pooling Convolutional Network for whole liver segmentation

By: Delmoral, J. Faria, D. Costa, D. Tavares, J.

Pyramid Dilated Residual Pooling Convolutional Network for whole liver segmentation

Jessica C. Delmoral[1]
jessica.delmoral@fe.up.pt
Diogo B. Faria[2]
dborgesfaria@gmail.com
Durval C. Costa[3]
durval.c.costa@gmail.com
João Manuel R. S. Tavares[1]
tavares@fe.up.pt

[1] Instituto de Ciência e Inovacão em Engenharia Mecânica e Industrial, Porto, Portugal
[2] School Of Health Sciences - University of Aveiro
[3] Champalimaud Centre for the Unknown, Fundação Champalimaud, Portugal

Abstract

Automatic liver segmentation in Computed Tomography (CT) images allows the extraction of the three-dimensional (3D) structure which is highly relevant in several clinical applications of diagnosis, surgical planning and disease surveillance. The adequate receptive field for the segmentation of such a big organ in CT images, from the remaining neighboring organs was very successfully improved by the use of the state-of-the-art Convolutional Neural Networks (CNN) algorithms, however, certain issues still arise and are highly dependent of pre- or post- processing methods to refine the final segmentations. Here, a Pyramid Dilated Residual Poling Convolutional Network (PDRP) is proposed, composed of an Encoder, a Dilation and a Decoder modules. The introduction of a dilation module has allowed the concatenation of feature maps with a richer contextual information. The hierarchical learning process of such feature maps, allows the decoder module of the model to have an improved capacity to analyze more internal pixel areas of the liver, with additional contextual information, given by different dilation convolutional layers. Experiments on the MICCAI Lits challenge dataset are described achieving segmentations with a mean Dice coefficient of 95.7%, using a total number 30 CT test volumes.

1 Introduction

Automatic segmentation of different medically relevant liver tissues is continuously an active research topic in medical image analysis. Such segmentations can provide doctors with meaningful and reliable quantitative information of the structure of the liver, which subsequently enable the identification of abnormalities. Knowledge of the liver structure becomes particularly relevant in individuals diagnosed with liver cancer. In this clinical scenario, physicians need to study the full liver physiology and make an informed decision on the treatment course. Liver cancer treatment may include chemo- or radio- therapy, hepatectomy (liver resection) or in very specific cases transplantation. Liver cancer has an alarming prevalence in a global scale and is the second most lethal cancer worldwide, accountable for more than 788,000 deaths in 2015 [6]. Computed Tomography (CT) is one of the most common modalities used for detection, diagnosis and follow-up of liver cancer [4]. Liver cancer is characterized by the development of abnormal cell accumulations, commonly referred as lesions that will appear represented differently within the anatomy of the liver, in structural images such as CT, and in many cases contiguous to the boundaries of the organ. Moreover, the appearance and shape variations of the liver is often significant among patients, the boundaries have limited contrast and are contiguous with neighbouring organs. Thus, in the clinical setting the image-aided diagnosis requires an accurate segmentation of the whole liver anatomy in CT images. In this paper, we present a new method for training a global CNN for liver segmentation in CT scans which addresses the issues above by developing a fully automatic liver segmentation model which efficiently combines the FCN with residual blocks and dilated convolutions.

2 Methods

2.1 Dataset

Public datasets are commonly used for assessing liver cancer in CT tissue segmentation algorithms as they provide ground truth labels. The model studied was trained and tested on data from the 2017 LITS MICCAI challenge dataset. We use a total of 130 CT volumes. Each image volume was characterized by a 512x512 image resolution and varying number of slices, ranging from 91 to 844. Having in mind the segmentation problem in this setting is regarded as a pixel-wise classification, the classification targets are comprised by two different classes: liver and background. To prevent extra errors induced by class data imbalances, the models proposed here were trained only on central slices depicting the liver area, comprising a total of 90 slices from each exam. CT image acquisition outputs a quantification of X-rays at a pixel wise level, which is outputted according to the known scale of Hounsfield units (HU), proportional to the degree of tissue attenuation suffered. Although different HU intervals characterize different organs, these values often overlap, making difficult the intuitive discrimination of the present tissues. To eliminate the noise effect of other HU value intervals, a technique named CT windowing is often applied. Thus, all CT slices were thresholded with a window range of [–200, 200] HU interval, recommended for liver analysis and to remove irrelevant tissue intensities.

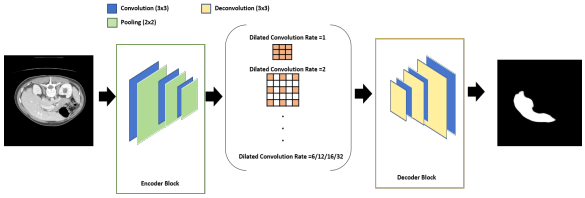


Figure 1: Neural network architecture of the Pyramid Dilated Residual Poling Convolutional Network (PDRP).

2.2 Feature learning of the proposed CNN model

Input images and the corresponding liver segmentation masks provided by human experts were used to train the network. Examples of ground truth masks are latter presented in the Results section. To learn the whole liver supervised features an FCN model was trained. Such model was formulated taking into account the several sizes of receptive fields that can allow the network to learn the most discriminative feature maps. Such methods require also the adequate kernelized image context to correctly identify the liver voxels. The size of receptive field roughly indicates the amount of context information that is used in each feature map.

The proposed architecture takes advantage of the dilated convolutions, introduced recently in CNN structures. The network works with 2D slice-wise axial images and is composed of (a) three encoding Residual Convolutional-Pooling blocks, followed by (b) five parallel layers of dilated convolutions with rate r = 1,2,4,6,8,16,32 which were concatenated and forwarded to (c) three deconvolutional deconding blocks. The network outputs are fine-tuned with a Sigmoid layer using the given labels. Such type of convolutional kernel is also rotation invariant. The dilations can be mentally conceptualized as the introduction of discrete intervals of pixel within the convolution kernel, that are dictated by the dilation rate r. Figure 1 illustrates the pipeline of the proposed training process.

2.3 Model training and parameter fine-tunning

The model parameters are learnt from training data by minimizing a loss function, via end-to-end training. Given the chosen dataset division a to-

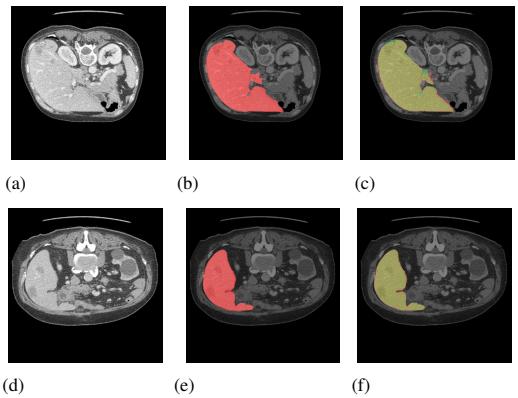


Figure 2: Examples of segmentation results of our proposed method. (a,d) represents the original CT, (b,e) the ground truth mask; and (c,f) the overlay of the obtained mask and the ground truth.

tal of 8100 image slices, from 100 subjects were used as training samples. For training we use the Adaptive Moment Estimation (Adam) algorithm [5] for model optimization during 70 epochs with learning rate reduced by a factor of 0.9 of the original value after 1/3 rd and 2/3 rd of the training finished. The network weight initialization with and without He norm initialization were explored. The dataset was augmented using rotation and horizontal flipping to increase generalizability of the model. The hyperparameters were tuned so as to give best performance on validation set. Training took 8 hours on one Nvidia Titan XP GPU. During training a set of 900 image slices were used as a validation dataset to monitor the models performance evolution during training. To deal with class distribution between the liver and background, the Dice Similarity Coefficient Loss was chosen for objective function minimization during model training. Such loss function has been extensively validated in the literature for Convolutional Neural Networks training, due its insensitivity to class imbalancing. The model inputs each image slice, of size 512x512 individually using only contextual information in the orthogonal direction. The model outputs the pixel-wise classification into each of the three classes.

3 Results

A total 2700 image slices, corresponding to a total of 30 3D CT scans, not previously used for model training, were used to test the performance of the proposed model. Table 1 shows the comparative test results of the proposed model and the top performing methods in the literature. To quantitatively evaluate the classification performance, we report the segmentation quality results of three metrics, proposed previously in the literature namely, the Dice (DSC) and Jaccard (JC) coefficients. The qualitative results of the segmentation results can be evaluated through Figure 2. In the two example results, the complex and heterogeneous structure of the liver were detected in the shown images. Overall, the model predictions were accurate in the classification of true positives. However, from the analysis of the entire dataset, the fuzyness of the liver boundaries in some scans leaked to the neighboring tissues, depicted in similar intensities. This is observable in Figure 2, in the example (c).

4 Discussion

In this work, we devise a simple, but efficient and automatic segmentation method, called Pyramid Dilated Residual Poling Convolutional Network (PDRP), that achieves state-of-the-art results in quantitative metrics when compared to the four top performing methods of the literature, as detailed in Table 1. To the best of our knowledge, no previous method taking advantage of the positive performance aspects of dilated convolutions was previously proposed for the task of liver segmentation in abdominal CT images. In medical imaging, the most traditional architecture for segmentation is the well-known U-Net, which is characterized by two distinct sequential blocks of encoder and decoder or contracting and expanding convolutional segments that basically aggregate semantic informations. We attempt to expand these feature with the addition of dropout, residual addition and dilated convolution operations. The simplicity of the proposed method when compared to some of the most traditional methods

Table 1: Liver segmentation performance results using different algorithms (HE - He weight initialization).

Method	Dice(%)	Jaccard (%)
[1]	89	
[3]	94.3	-
[2]	95.90	92.19
[7]	96.3	-
Ours	93.1	89.4
Ours w/ HN	95.7	91.3

used such as the U-Net [3], provided 1) better performing results, but also 2) a parameter reduction that is achieved by the efficiency of the inclusion of the dilated convolutions.

5 Conclusions

In the present work, a novel CNN architecture for whole liver segmentation in CT images is proposed. The segmentation of a big organ such as the liver, would in many previous architectures be penalized by an inadequacy of the receptive field used for feature learning in previously proposed architectures. The key concatenation of dilation convolutions has allowed accurate segmentations of the final liver boundaries, with minimal fuzziness. No hole filling post-processing was needed with the proposed architecture. In future works, the proposed architecture potential to segment other liver tissues, such as lesions and vascular structure will be explored. Moreover, advanced techniques of data augmentation using adversarial networks, could further improve the resulting segmentations obtained in the present study.

Acknowledgments.

The authors gratefully acknowledge the funding from Project NORTE-01-0145-FEDER- 000022 - SciTech - Science and Technology for Competitive and Sustainable Industries, cofinanced by "Programa Operacional Regional do Norte" (NORTE2020), through "Fundo Europeu de Desenvolvimento Regiona" (FEDER). The authors also kindly thank Nvidia, for the contribution with one Nvidia Titan XP GPU, that was used in this work.

References

[1] A Ben-Cohen, I. Diamant, E. Klang, M. Amitai, and H. Greenspan. Fully convolutional network for liver segmentation and lesions detection. In G et. al Carneiro, editor, *Deep Learning and Data Labeling for Medical Applications*, pages 77–85, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46976-8.

[2] Lei Bi, Jinman Kim, Ashnil Kumar, and Dagan Feng. Automatic liver lesion detection using cascaded deep residual networks. 2017. URL `http://arxiv.org/abs/1704.02703`.

[3] P. F. Christ et al. Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks. *arXiv preprint arXiv:1702.05970*, 2017.

[4] Lucy E. Hann, Corinne B. Winston, Karen T. Brown, and Timothy Akhurst. Diagnostic imaging approaches and relationship to hepatobiliary cancer staging and therapy. *Seminars in Surgical Oncology*, 19(2):94–115, 2000. ISSN 1098-2388.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] World Health Organization. Fact sheet: cancer, 2015. URL `http://www.who.int/mediacentre/factsheets/fs297/en/`.

[7] Yading Yuan. Hierarchical convolutional-deconvolutional neural networks for automatic liver and tumor segmentation. 2017. URL `http://arxiv.org/abs/1704.02703`.

Paper:#36

Camera Adaptation for Deep Depth from Light Fields

By: Portela, D. Monteiro, N. Gaspar, J.

Camera Adaptation for Deep Depth from Light Fields

Diogo Filipe Baptista Portela
diogo.b.portela@gmail.com
Nuno Barroso Monteiro
nmonteiro@isr.tecnico.ulisboa.pt
José António Gaspar
jag@isr.tecnico.ulisboa.pt

Institute for Systems and Robotics
Instituto Superior Técnico
University of Lisbon, Portugal

Abstract

Plenoptic cameras image a 3D point by discriminating light rays contributions towards various viewpoints. They allow developing depth estimation methods, such as depth from focus as found in the deep neural network DDFFNet by Hazirbas et al. The training of the DDFFNet has implicit a specific camera geometry, defined by the microlens array and the configuration (zoom and focusing) of the main lens. In this paper we augment the network application range by accepting larger input disparity ranges that can be obtained by different configurations or cameras. The proposed methodology involves converting a field of view and a depth range into the settings where the DDFFNet has been trained. The conversion of the input data is based in the estimation of gradients (structure tensor) on the light field. Results show that depth estimation is possible for various cameras while using the originally trained DDFFNet.

1 Introduction

The last years have seen a rise in the study and improvement of pleonoptic cameras since the first model was developed by Ng [4], in 2006. The use of these cameras allow to obtain, from a single shot, what is called a light field, an array of multiple scene views named viewpoints, as if an array of conventional pinhole cameras were used. A light field can be digitally refocused after it has been captured, as demonstrated by Ng *et al*. [3], and used to achieve depth reconstruction, as shown by Tao *et al*. [5].

Hazirbas *et al*. [1] presented *Deep Depth From Focus Network* (DDFFNet), a Convolutional Neural Network that outputs a disparity map from a focal stack. As any neural network it requires intense training, and while it may lead to good test results, it may also result in an inability to perform well under inputs with characteristics outside its training scope. In this paper we deal with the network's inability to correctly reconstruct datasets with disparity ranges outside its scope. These ranges can vary widely depending either on the camera's zoom or focus or its physical characteristics, such its baseline.

Although the usual approach to solve this problem lying on fine tuning, that is not always possible when dealing with new data, due to constrains such as few number of examples, time or computational power.

The method presented here tries to enlarge the application range by obtaining a new light field by backprojecting the original, transforming it and finally reprojecting the result into an array of cameras identical to Hazirbas'.

2 From light fields to the Deep Depth From Focus Network

We can describe a light ray using the pixel it hits, in the form of a light field $\mathcal{L}(i,j,k,l)$, where (k,l) indicates the viewpoint's index within the array, while (k,l) indicates the pixel in the viewpoint.

We focus on a disparity $\frac{\partial i}{\partial k} = \frac{\partial j}{\partial l} = \alpha$ by performing shearing, this is translating each viewpoint by an amount proportional to its distance from the array's center, $\mathcal{L}_\alpha(i,j,k_\alpha,l_\alpha) = \mathcal{L}(i,j,k+\alpha(i_{center}-i),l+\alpha(j_{center}-j))$, and summing them. By stacking multiple images, each focused at a different disparity, we obtain what is called a focal stack.

The DDFFNet takes a focal stack of 10 images, each focused at linearly spaced disparities, to produce a disparity map as output. The network was trained for a depth range of [0.5, 7] meters, meaning, by the camera parameters, that the input focal stacks cover disparities between [0.02, 0.28] pixels. Datasets with ranges outside this scope are incorrectly reconstructed. Retraining is not always possible, so we propose simulating a camera as the one used in the training process by transforming the new light field in one similar to the ones captured for training the DDFFNet.

3 Ground Truth based Camera Adaptation

Considering a plenoptic camera as an array of pinhole cameras, its field of view can be bounded by the envelope of all cameras' fields of view (pyramids). This envelope is not much wider than the central viewpoint's pyramid, because usually baselines are very small.

We will transform the new dataset, as a point cloud, from the original trunk of pyramid to one similar to the DDFFNet camera, forming a new light field similar to the one the latter would captured. As example, we used the dataset *Cotton*, Figure 1(f), present in the 4D Light Field Benchmark [2], which its disparity range is outside the ones used for training the network.

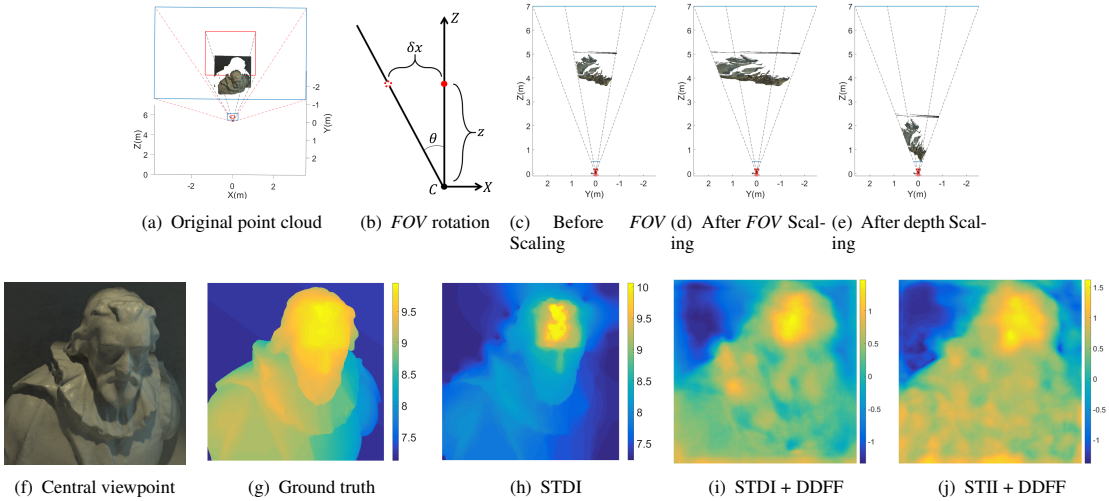
Backprojection In the first step we backproject the center viewpoint, resulting in a 3D point cloud, as in Figure 1(a). Besides the camera's intrinsic parameters, we need a depth estimation for each pixel, obtain through the ground truth or through some depth estimation method, such as the structure tensor, explained in detail in section 4. With depth Z we compute the other 3D coordinates, (X,Y), using the backprojection model $[X\ Y\ Z\ 1]^T = [C^T\ 1]^T + [Z.D^T\ 0]^T$ where $C = -P_{1:3}^{-1} \cdot P_4$ represents the optical center, $D = P_{1:3}^{-1} \cdot [u\quad v\quad 1]^T$ is the optical ray's direction for a given (u,v) pixel and P_i the projection matrix's i^{th} column.

***FOV* rotation and scaling** We obtain the two model's fields of view by backprojecting the image corners, Figure 1(a) with Benchmark and Hazirbas' in red and blue, respectively. To align their centers, the point cloud is rotated along X and Y around the optical center. For each, the rotation angle can be computed backprojecting both cameras' principal points to a depth z, Figure 1(b) where the red and white dot are the Benchmark and Hazirbas' projection, respectively. We conclude that $\theta = tan^{-1}(\delta x/z)$. The other dimension's angle can be computed in an analogous form.

To match the field of views vertex angles, we scaled X and Y, by the same factor to avoid distortion. However, due to their different shapes (Benchmark's is a square while Hazirbas' a rectangle), the scaling factor is the one that scales the point cloud so that it matches Hazirbas' smaller side, that is, the ratio between their maximum Y, for the same depth. The result of this scaling can be visualized in Figures 1(c) and 1(d).

Depth scaling We have now to scale the point cloud so that its depth lies in network's trained range. However, inspecting Figure 2(b) of the supplementary material available in [1], we conclude that using a smaller range will result in a more well distributed set of focused depths. Thus the range used was [0.5, 2.5] meters. Through an affine transformation we can force the depth to fall within a range $[z_1,z_2]$ by solving the linear system, $z_{min}a + b = z_1$ and $z_{max}a + b = z_2$, with z_{max}, z_{min} the original maximum and minimum depths, respectively. To compensate for this, the X and Y are multiplied, $(x_{new},y_{new}) = \frac{z_{new}}{z} \cdot (x,y)$, for each point. See Figure 1(e).

Reprojection The final step is to project the point cloud to a camera array with the same intrinsic and extrinsic parameters as the camera used in the DDFFNet. This results in the new light field which will then be refocused and used to create the input focal stack.



(a) Original point cloud (b) *FOV* rotation (c) Before *FOV* Scaling (d) After *FOV* Scaling (e) After depth Scaling

(f) Central viewpoint (g) Ground truth (h) STDI (i) STDI + DDFF (j) STII + DDFF

Figure 1: Depth reconstruction from a light field. Light field central viewpoint (f) and ground truth data (a, g) from the dataset [2]. Ground truth based camera adaptation, point cloud transformations (b - e). Camera adaptation based on the structure tensor (h - j).

4 Structure Tensor based Camera Adaptation

In real cases 3D point clouds are not available. We propose obtaining an initial depth estimation to construct the point cloud.

In a light field, slices made by fixing (i,k) or (j,l) will result in an epipolar image. The disparity of a feature will translate as the slope of a epipolar line in these images, being parallel to the gradient direction such that $\frac{\partial i}{\partial k} = -\frac{\nabla_i \mathcal{L}}{\nabla_k \mathcal{L}}$, as in Figure 2. By measuring the gradient in those images we can extract a depth estimation.

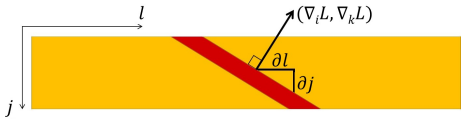


Figure 2: Epipolar plane. Depth information can be obtained from the gradient.

For this we need to obtain each pixel structure tensor, $S(k,l)$, a matrix derived from the gradient that will give its predominant direction in that pixel.

Let $I^{ik}_{(.)}(j,l)$ be the value of $\nabla_{(.)}L$ calculated at (j,l), for a horizontal epipolar image, calculated using a Sobel operator. For each pixel the local structure tensor, $S_0(j,l)$, is computed as

$$S_0(j,l) = \begin{bmatrix} I^{ik}_j(j,l)^2 & I^{ik}_j(j,l)I^{ik}_l(j,l) \\ I^{ik}_l(j,l)I^{ik}_j(j,l) & I^{ik}_l(j,l)^2 \end{bmatrix}. \quad (1)$$

Values are then averaged along j, resulting in a 1D array $S^{ik}_e(l)$. This process is repeated for every horizontal and vertical epipolar image.

By computing $S(k,l) = \sum_i \sum_j S^{ik}_e(l) + S^{jl}_e(k)$, where $S^{jl}_e(k)$ represents the average 1D array for vertical epipolar images, we obtain the value of the final structure tensor.

In a structure tensor matrix, computing the eigenvector corresponding to the greatest eigenvalue, λ_1, gives the predominant gradient direction. The relation between both eigenvalues allow for a confidence level on the gradient obtained. Such measure was defined as $\lambda_1 - \lambda_2$, with cases below a given threshold discarded.

After computing the structure array, its eigenvectors are calculated and filtered, and from them the gradient directions are computed. These are then used to compute the disparity for each pixel. This disparity map is converted to depth and used to construct the point cloud.

To deal with areas of low gradient being discarded, we propose two strategies. Constructing the point cloud as is and inpaint each viewpoint in intensities, or inpainting the disparity map, and then projecting a full point cloud.

5 Experimental results

Given the focal stack input, the network was used to obtained a new point cloud to be transformed using the inverse transformation of each step, in reverse order. Each point is projected to a camera identical to the Benchmark central viewpoint, forming a depth map, then converted to disparity and compared to the ground truth.

The proposed method was evaluated on the Benchmark's [2] Training Set, the same used by Hazirbas to evaluate the network performance after retraining. In Table 1 are the numerical results, as defined in [1]. Pre refer to results using the untransformed datasets. GT concerns the ground truth based approach. STDI and STII refer to structure tensor methods complemented with disparity or intensity inpainting, respectively. As a qualitative analysis, the disparity map obtained by each method is presented in Figure 1, along with the ground truth and central viewpoint.

Method	Pre	Retrain	GT	STDI	STII+DDFF	STDI+DDFF
Disparity MSE	0.7741	0.19	0.3002	0.7383	0.5378	0.3392
Disparity RMS	0.8709	0.42	0.5463	0.7934	0.7227	0.5765
Depth MSE	0.9395	—	0.2934	1.1063	0.6499	0.3104
Depth RMS	0.7233	—	0.4220	0.6950	0.5958	0.4332

Table 1: Retrain and ground truth (GT) vs structure tensor methods.

Analyzing the results we conclude that applying the network directly on the 4D Benchmark datasets produces an MSE in depth of almost 1 meter, rendering it almost useless. However, by applying the proposed method through the Structure Tensor + Disparity Inpainting technique we reduce that error by more than two thirds (≈67%).

6 Conclusions

In this paper we presented a method to overcome the small disparity range limitation of the DDFFNet without resorting to retrain. The datasets used in this proof of concept were restricted to the benchmark [2], however, other datasets can be transformed to a valid DDFFNet input, provided the intrinsic parameters of the camera are known. Comparing to a full retraining approach, the proposed method provides a faster, more versatile and adapting approach at the cost of loosing some accuracy.

Acknowledgements

Work partially supported by the FCT project UID / EEA / 50009 / 2013.

References

[1] C. Hazirbas, L. Leal-Taixé, and D. Cramers. Deep depth from focus, November 2017. arXiv:1704.01085v2.

[2] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *ACCV*, 2016.

[3] R. Ng. Fourier slice photography. *ACM Transactions on Graphics*, 24:735–744, 2005.

[4] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Comput. Sci. Dept., Stanford Univ., Tech. Rep.*, 2004.

[5] M. Tao, P. Srinivasa, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. *CVPR*, June 2015.

Paper:#37

An Acquisition System for Electrodermal Activity Signals Used to Identify Skin Conductance Patterns Associated with Human Emotional States

By: Lopes, F. Fonseca, I. Azevedo, A. Gomes, V.

An Acquisition System for Electrodermal Activity Signals Used to Identify Skin Conductance Patterns Associated with Human Emotional States

Adriana Azevedo*,1
Viviana Gomes*,1
Inácio Fonseca[1]
Fernando Lopes[2]

[1]Coimbra Institute of Engineering, Polytechnic Institute of Coimbra

[2]Telecommunications Institute - Coimbra

Abstract

This paper describes an Acquisition System for signals representing the human ElectroDermal Activity (EDA). The developed system is composed of a hardware component and a software interface that allow to acquire, process and save a voltage signal representing the Skin Conductance Response (SCR) of a human subject. A specific design objective was the ability to use the standard soundcard input of a personal computer and thus avoid extra complexity for the acquisition hardware. The EDA is controlled by the autonomic nervous system in the human body. This organ property can be measured by variations in the conductance of the skin, which can in turn be related to the psychological and emotional state of the person [1]. We tested our acquisition system using a small set of five subjects when viewing relaxing videos interrupted by unexpected horror and accident scenes. Our final objective is to be able to monitor the evolution of the emotional state in the educational environment and act to maximize the study efficiency. For this purpose, the software must evolve to classify typical response patterns. This paper focus on the first part - the hardware and software acquisition system.

1 Introduction

Several medical experiments have shown that the magnitude of the electrical conductance in a person's skin is correlated with central psychological phenomena and thus can be related to the person's emotional state (Figure 1). As a result, a change in the emotional state can be observed by a change in the skin conductance, allowing to quantify the level of arousal of the person. One of the most common examples where the ElectroDermal Activity of the skin (EDA) is used to infer the emotional state, is through the use of the polygraph. The EDA corresponds to a transient change in the skin properties resulting from sweat secretion and sweat gland activity [2]. In the case of sweat excretion this causes changes in the conductance of the skin surface. Sweat gland activity originate in the sympathetic subdivision of the Autonomic Nervous System (ANS) [2]. The Autonomic Nervous System is the part of the nervous system that regulates the functions of some internal organs, without interference of the will, as is the case of breathing, blood circulation, temperature control and digestion.

The EDA can be sensed through several techniques, including the direct variation of the skin resistance or skin conductivity, that is considered in the present work. An example of a commercial system for EDA acquisition and analysis is the one from Affectiva, that presents a system that incorporates the skin conductance signals with other biological signals to detect car driver emotions. This system uses sensors such as the E4 Wristband from Empatica [3,4].

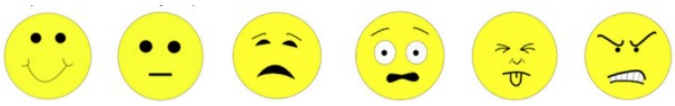


Figure 1: Possible human emotional states.

2 Concepts

The conductivity of the skin can be characterized in two different types, as illustrated in Figure 2: Tonic Conductivity of the skin, which refers to the base level of the skin conductance, and is generally referred to as Skin Conductance Level (SCL); and Skin Conductance Response (SCR), which is the type of conductance that changes its value with a frequency between 0.05 - 1Hz when an event occurs. A typical skin conductance response can be between 0.002 - 1 µS (microsiemens) [5].

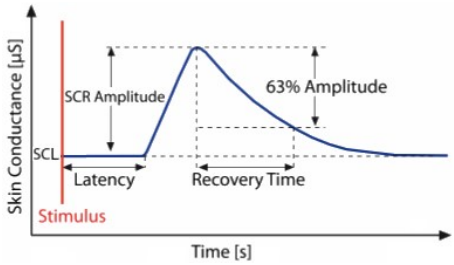


Figure 2: Skin conductance variation when there is an emotional stimulus [5].

For the best results the EDA is usually measured in parts of the skin where the density of sweat glands is greatest. Therefore, the most common arrangement of electrodes is in the palm eminences of the hand, the distal phalanges or the middle phalanges, as shown in Figure 3. These are the body parts that best conjugate the desired properties, with accessibility and stability for the placement of the electrodes. As a reference one may use a part of the skin with fewer or no sweat glands, or a part in the same area as the active electrode [5].

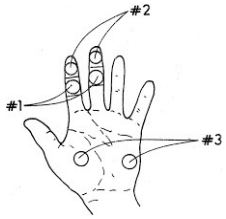


Figure 3: Three most common provisions for measuring EDA [5].

When the body is under stress the production of sweat by the sweat glands increases. Sweat is a weak electrolyte and a good conductor. Increased sweat production on the skin creates several low resistance pathways through its surface. A relaxed individual who has dry skin will have a high electrical resistance, while a subject under stress will produce more sweat, thus will have a lower electrical skin resistance. Since the resistance is the inverse of the conductance, we can say that stress influences the Skin Conductance, G, that is expressed in Siemens (S), with the skin resistance expressed in Ohm (Ω) (Equation 1) [6].

$$G[\mu S] = 1/R[k\Omega] \quad (1)$$

3 Acquisition Hardware and Software

The acquisition hardware circuit is divided into three main parts: a transconductance amplifier feeding a Voltage Controlled Oscillator (VCO), a low-pass filter (Sallen-Key topology) and an opto-isolator audio interface. The block diagram and electrical schematic can be observed in Figure 4 and Figure 5 respectively. The first stage of the amplifier is based in a circuit with a similar objective in [5].

The amplifier converts the skin resistance to a voltage value that is low-pass filtered at 0.5 Hz. The filtered voltage drives a VCO that generates a frequency modulated pulse train that is then passed through a 5 kV opto-isolator for safe galvanic isolation. The frequency modulated pulse train is further low-pass filtered and input to the computer soundcard as an analog audio signal (Figure 6).

* Licenciatura Degree in Biomedical Engineering Student - ISEC

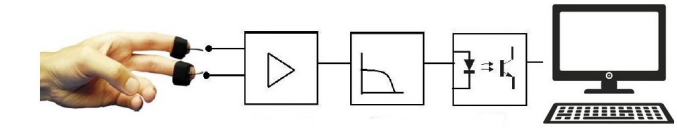


Figure 4: Block diagram of the acquisition system.

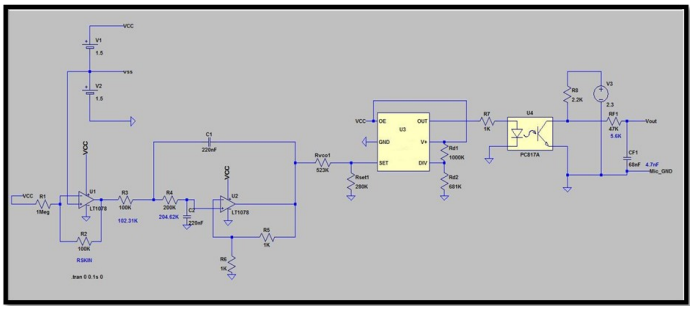


Figure 5: Circuit schematic for LTSpice simulation (implemented).

The software application was developed in Matlab and performs the real-time sound acquisition and frequency demodulation (see Figure 7). The acquisition can be continuous or time-limited. During acquisition and demodulation, the EDA signal is displayed in real-time and a complete session for a given subject can be saved for later display and analysis or for archiving purposes. The EDA signal can be processed for pattern recognition and graphical interpretation.

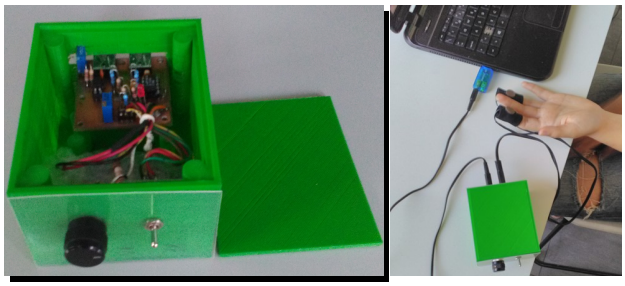


Figure 6: Left: 3D printed Box and built acquisition board. Right: System being used in acquiring EDA signals.

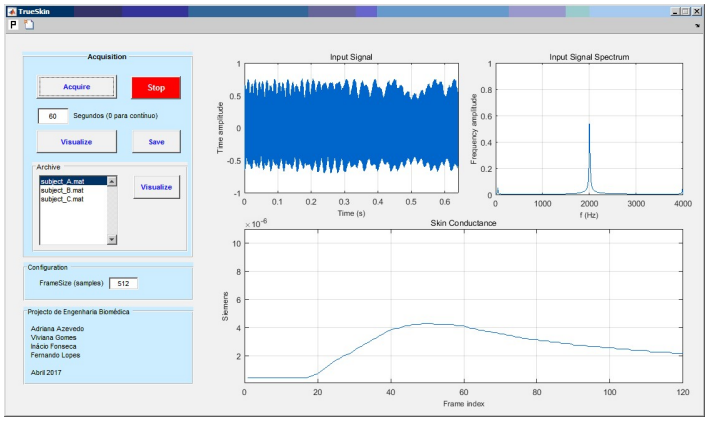


Figure 7: Matlab Application. Serves as User Interface and demodulates and processes the signal acquired through the sound card.

4 Validation of the Acquisition System

To validate the developed acquisition system an experimental setup was created. A three-minute video was composed consisting in relaxing landscapes and music. Two small clips, one with a scary face and one with a very serious road accident (with accompanying sound) were inserted in two random locations. A set of five subjects were invited to visualize the video without prior knowledge of its contents. Figure 8 shows the observed relationship between the scenes present in the video, including the "stressing" moments, and the corresponding measured skin conductance patterns. In the same way, the signal present in Figure 6 was observed in an independent test, where a subject watched a different video that included a serious motorcycle accident. In general, results vary from individual to individual and also vary for the same individual if the sequence is viewed more than once (surprise effect lost).

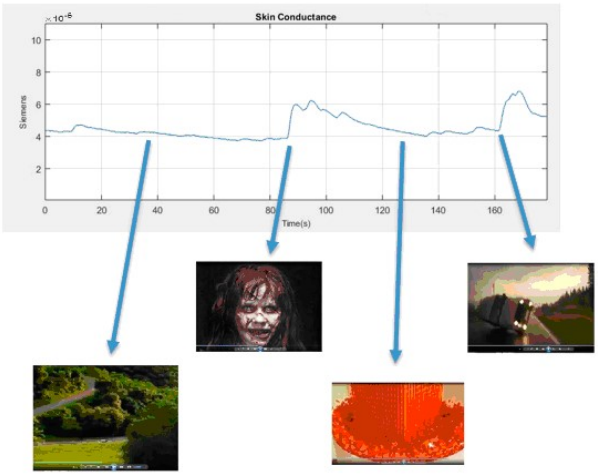


Figure 8: Skin conductance signal and video contents.

In Table 1 we present the calculated time and amplitude features from the EDA signals associated with each subject, as defined in Figure 1. Different features point to different emotional responses among the various individuals.

Table 1: Results obtained for five different individuals.

Individual	Latency Time (s)	Rise Time (s)	Half-Recovery Time (s)	Amplitude (μS)
A	2,78	14,08	14,64	2,33
B	2,79	5,37	11,62	4,61
C	1,87	7,42	18,27	6,27
D	1,41	18,18	15,50	1,41
E	1,98	7,68	5,38	4,47

5 Conclusions and Future Work

In this paper we presented the development of the hardware and software components of an acquisition system for EDA signals. A very unique characteristic of this system is that it includes a frequency modulation step that allows the use of a common computer soundcard as input interface. The performed tests demonstrated that the developed system works very well and allows to measure the EDA signal in a very precise and robust manner, including adaptation to different average levels of sweat. Future work will compare the acquisition performance with established equipment and will also involve obtaining signal patterns and features for specific pre-determined emotions, allowing to create reference and test patterns for emotional state classification. Artefacts such as those arising from motion and respiration need to be considered. Our long term objective is the study of the evolution of the emotional states of a studying individual.

References

[1] Rosalind W. Picard. *Toward a wearable autonomic sensor*. Sc.D., FIEEE, Affective Computing at MIT Media Lab [Online]. Available: https://affect.media.mit.edu/projectpages/iCalm/iCalm-2-Q.html

[2] Raphaela Schnittker. *Eletrodermal activity of novice drivers during driving Simulator training - an Explorative study*, [Online]. Available: http://essay.utwente.nl/61873/1/Schnittker, R. - s1016350_(verslag).pdf.

[3] Affectiva, Inc. [Online]. Available: www.affetiva.com

[4] Empatica, Inc. *E4 Wristband - watch that gets biological signals on the wrist*, [Online]. Available: https://www.empatica.com/en-eu/research/e4/.

[5] José Guerreiro. *A Biosignal Embedded System for Physiological Computing*. [Online]. Available: https://www.researchgate.net/publication/273448313_A_Biosignal_Embedded_System_for_Physiological_Computing.

[6] Michael E. Dawson, Anne M. Schell, Diane L. Filion e Gary G. Berntson. *The Eletrodermal System*. [Online]. Available: http://dornsife.usc.edu/assets/sites/585/docs/handbookchapter2000.pdf.

Paper:#45

Analysis of Autoencoders for Feature Representation of Protein Sequences

By: Albuquerque, J. Pereira, C. Arrais, J.

Analysis of Autoencoders for Feature Representation of Protein Sequences

João Albuquerque[1, 3]

Carlos Pereira[1, 2]

Joel P. Arrais[1]

[1] CISUC, University of Coimbra, Department of Informatics (DEI)

[2] Polytechnic Institute of Coimbra, Coimbra Institute of Engineering, IPC/ISEC

[3] Department of Physics, Coimbra University

Abstract

Although proteins participate in virtually all biological processes in the organism, for most, their structure and functions are yet to be experimentally validated. The ability to apply computational models to determine its function from the primary structure has proven of major relevance. Despite multiple contributions towards this problem, recent works using Artificial Neural Networks have shown major improvements over the state of the art. One open problem that persists consists in getting the optimal feature representation from a given amino acids sequence. In this work we explore the use of autoencoders to find the optimal encoding providing the base for dimension reduction. We evaluate the influence of different parameters as well as different feature sets to achieve an improved representation of the original input space. The obtained results of a Linear SVM classifier show an improvement of the compacted representation in the classification of a set of protein function.

Keywords: Protein Classification, Feature Extraction, Autoencoders, SVM.

1 Introduction

Proteins are the backbone of most part of biological functions. Their action is mainly determined by their folding and sequencing. Therefore, the ability to predict its structure represents an invaluable advance in predicting its function in the organism [1,2].

Autoencoders are one of the types of Artificial Neural Networks (ANN) now also being used in the field of computational biology. With the expansion of Big Biological Data, and especially with the completion of the Human Genome Project [3], the problem of protein classification and feature extraction gained a new importance.

Concerning the dimensionality reduction of the protein representation, several techniques are used, such as Principal Components Analysis, Linear Discriminant Analysis and more recently, Autoencoders. The Autoencoder has been in several domains, proving its effectiveness also in Biological field, namely Protein-Protein interaction detection [4,5] and prediction of protein structural classes [1].

This work focuses on the analysis of Autoencoders. The main goal is to discover the influence that the representation of the protein sequence in partial segments and the Autoencoder parameters, like activation transfer function and size of the hidden layer, have in the methods capability to effectively reduce the dimension of the features input space, evaluated by a SVM classifier of proteins in two different families.

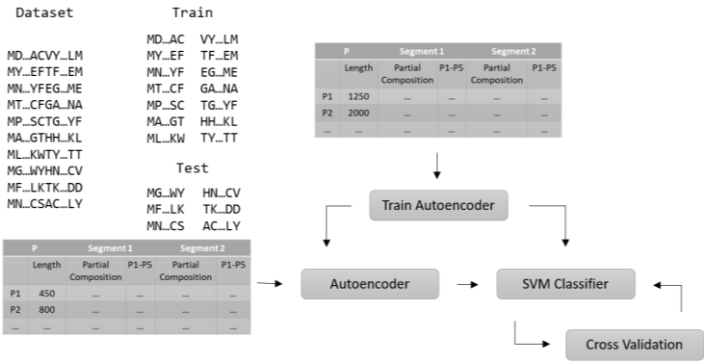


Figure 1: Structure of problem. The autoencoder learns the compacted features representation from training data (protein sequences) then given to the SVM for protein classification.

2 Protein Sequences Feature Extraction

Each protein is a sequence of amino acids, a result from the construction of a chain in the cytoplasm of each cell. This sequence, in its final form, varies in length, composition, physicochemical properties and function. Each organism has a set of proteins, different or more alike from other organisms, and each protein belongs to certain families.

Taking that sequence, we can then divide it into a fixed number of segments, evaluate its partial composition and together with the chemical properties of each amino acid, come up with a set of features to be encoded, as exhibited in Figure 1.

3 Autoencoders

Autoencoders are unsupervised ANNs used normally for the reduction of a feature input space, whether it consists of an image or a sequence. The goal is to provide a hidden representation of the input space, and then decode the information obtained in the compressed space, obtaining an output as similar as possible to the original set.

The chosen layer is known as “the bottleneck” or code layer, in which maximum reduction is achieved through encoding, and reliable output is obtained through decoding.

The difference between the output and input is the target to minimize, being more frequently used the minimum squared distances – L(input, output). The method implementation is based on the MatLab ANN Toolbox [6].

4 Experiments and Dataset

Concerning the dataset, the yeast has been the selected organism to form the training and testing datasets of protein sequences in the Autoencoder phase, using the proportion of 70% and 30% respectively (Table 1), from the original UniProt Dataset [7]. Another group of proteins was selected, from two different families – Ribossomal and Transferase Hexapeptidase – to assess the quality of the Autoencoder with an associated SVM, in the classification stage (Table 2).

Organism	Training	Testing
Yeast	2371	711

Table 1: Test and training datasets for the autoencoder.

Family	Training	Testing
Transferase Hexapeptide	732	315
Ribossomal	698	300

Table 2: Test and training datasets for the classifier.

For different number of protein segments, a dataset was formed, as also exemplified in Figure 1. The extracted features include the overall length of the sequence and, for every segment, the partial composition and a set of physicochemical properties – Hydrophobicity, Normalized Van der Waals Volume, Polarity, Polarizability, Secondary structure and Solvent-accessibility [8] – has been taken.

As the number of features increases with the number of protein segments, a fixed hidden size in the autoencoder would lead to higher MSE for larger dimensions. To minimize this problem, the parameter of hidden size is defined as a percentage of the original input space.

In the experimental setup, the parameters tuned were the hidden size, the encoder and decoder transfer functions and sparsity proportion, as well as the number of sequence segments.

For each autoencoder configuration, 15 runs were made keeping the average and the standard deviation of MSE.

In the final stage, an SVM classifier was applied to determine if the information obtained through an autoencoder is relevant for the classification of proteins.

5 Results and Discussion

The output dataset is constructed with the autoencoder prediction function. By comparison of the input and output data, the results shown in Figures 2 and 3 and Tables 3 and 4 were obtained.

Function	Log(MSE)	Log(STD)
Saturation Linear	2.6044	2.9505
Logarithmic Sigmoid	3.2428	3.0680

Table 3: Mean squared error regarding the use of both Transfer Function as encoder function.

Function	Log(MSE)	Log(STD)
Saturation Linear	4.4509	3.5604
Logarithmic Sigmoid	1.3963	0.9576

Table 4: Mean squared error regarding the use of both Transfer Functions as decoder function.

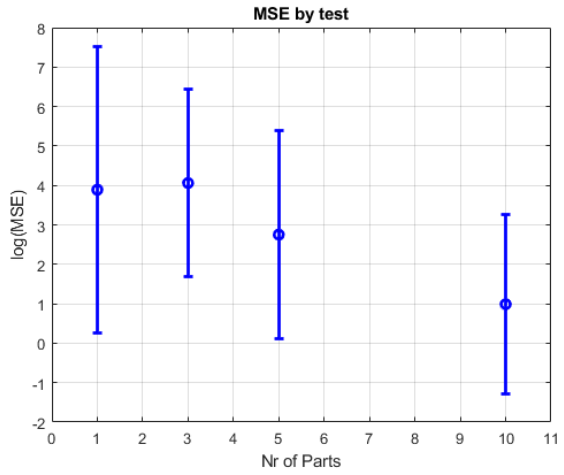


Figure 2: Variation of the mean squared error with the number of parts each sequence was divided (1,3,5 and 10 segments).

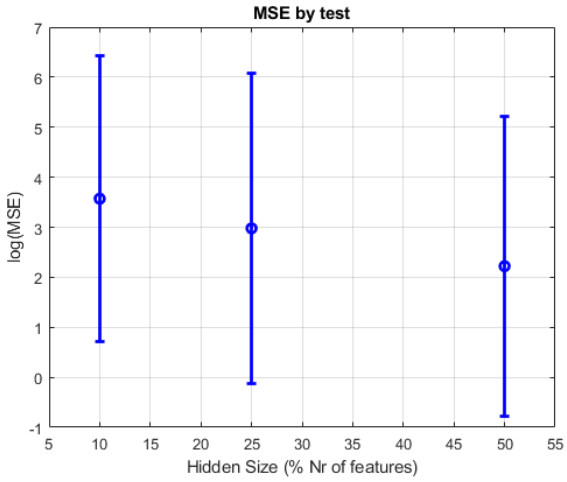


Figure 3: Variation of the mean squared error with the hidden size of the Autoencoder (10%, 25% and 50% of the number of features).

These results illustrate how the parameters affect the performance of the autoencoder.

Regarding the selection of the transfer functions, Table 3 presents a small difference in the mean values of MSE considering the standard deviation. Therefore, any extrapolation must be taken as only a reference, and not as a real evidence.

The same does not happen in Table 4, showing that the Logarithmic Sigmoid is a more reliable decoder transfer function for this case study.

Figure 2 shows a decrease of the error while the number of segments grows. Although the total number of features is higher, as each part of the sequence may contain specific information, this helps to find most relevant hidden representations. Future work will apply the reconnaissance of motifs in the sequence according to its biological function, possibly allowing for even better and significant results.

Figure 3 shows that the growing hidden size, as percentage of the size of the input, increases the precision of the autoencoder.

Applying a Support Vector Machine to the second dataset with the purpose of classifying proteins according to their family, the Table 5 shows the achieved results regarding the Area Under the Curve (AUC) criterium.

Parameter		Mean(AUC)	Std(AUC)
Number of Segments	1	0.95448	0.032848
	3	0.95861	0.080535
	5	0.97278	0.030883
	10	0.98134	0.022452
Percentage of Hidden Size	10	0.96872	0.030625
	25	0.96410	0.073058
	50	0.96759	0.028085
Encoder Transfer Function	Logarithmic Sigmoid	0.98575	0.022314
	Saturation Linear	0.94785	0.058804
Decoder Transfer Function	Logarithmic Sigmoid	0.96645	0.029661
	Saturation Linear	0.96716	0.061677

Table 5: Mean and standard deviation of the Area Under the Curve of the Classifier with the use of different parameters in the Autoencoder.

Due to the high mean AUC values, we can conclude that the quality of the representation is not lost in the hidden representation of the autoencoder but can be optimized by a suitable parameters fine tuning. The number of segments and the encoder transfer function appear as the most relevant parameters for the classifier performance.

6 Conclusion

This work achieved with relative success the goal of proving that the autoencoder indeed allows a specific set of features to be reduced in dimension without losing its information quality. However, no configuration is optimal for all datasets and for each problem a specific parameters setting must be done to achieve good performance.

The division of the sequence in several segments increase the input space dimension but helps to improve the classifier precision. While the computational power may be of higher demand, the precision will most likely increase if the search for known motifs is pursued. This search must not be exclusive, and original features should also include the whole sequence, giving the ANN the possibility to discover unknown relation among data.

The quality of the reduced representation of the feature space is confirmed by the SVM regarding protein function identification, and might be used in future work, applied to the problem of drug target discovery [9].

Acknowledgments

This research has been funded by the Portuguese Research Agency FCT, through D4 - Deep Drug Discovery and Deployment (CENTRO-01-0145-FEDER-029266).

References

[1] Sam Sinai and Eric Kelsic. *Using a Variational Auto-encoder to predict protein function*, 2017

[2] Liu Jian-wei, Chi Guang-hui, Liu Ze-yu,Liu Yuan,Li Hai-en, Luo Xiong-Lin. *Predicting Protein Structural Classes with Autoencoder Neural Networks*, 2013

[3] Francis S. Collins, Michael Morgan, Aristides Patrinos. *The Human Genome Project: Lessons from Large-Scale Biology*, 2003

[4] Edgar D. Coelho, Igor N. Cruz, André Santiago, José Luis Oliveira, António Dourado and Joel P. Arrais. *A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions*, 2017

[5] Wang YB1, You ZH, Li X, Jiang TH, Chen X, Zhou X, Wang L. *Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network*, 2017

[6] M. H. Beale, M. T. Hagan and H. B. Demuth. *Neural Network Toolbox™ User's Guide*, 2018.

[7] The UniProt Consortium. *UniProt: the universal protein knowledgebase Nucleic Acids Res. 45: D158-D169*, 2017

[8] Chang Zhou, Hua Yu, Yijie Ding, Fei Guo, Xiu-Jun Gong. *Multi-scale encoding of amino acid sequences for predicting protein interactions using gradient boosting decision tree*, 2017

[9] Edgar D. Coelho, Joel P. Arrais, José Luís Oliveira. *Computational Discovery of Putative Leads for Drug Repositioning through Drug-Target Interaction Prediction*, 2017

Poster Session 2

Genome/Drugs, Physical/HW Systems and Methods

Paper:#2

Mobile Human Shape Superimposition using OpenPose: An Initial Approach

By: Bajireanu, R. Veiga, R. Pereira, J. Sardo, J. Lam, R. Cardoso, P. Rodrigues, J.

Mobile Human Shape Superimposition using OpenPose: An Initial Approach

Bajireanu, Roman
romanbajireanu@gmail.com
Veiga, Ricardo J.M.
ricardojorge@martinsveiga.com
Pereira, João A.R.
japereira@ualg.pt
Sardo, João D.P.
joao_dps@outlook.com
Lam, Roberto
http://w3.ualg.pt/~rlam/
Cardoso, Pedro J.S.
http://w3.ualg.pt/~pcardoso/
Rodrigues, João M.F.
http://w3.ualg.pt/~jrodrig/

LARSyS & Instituto Superior de Engenharia
University of the Algarve
Faro, Portugal

Abstract

To improve user's museum experiences a mobile Augmented Reality (AR) framework is being developed, as a part of the Mobile Five Senses Augmented Reality (M5SAR) system for museums project. This paper presents an initial approach to develop one module of this framework: the human shape detection and content superimposition module. The human body joints information and texture overlapping is used to achieve the goal. OpenPose model was used to detect human body joints. At this point, the initial results and proof-of-concept are presented.

1 Introduction

This work presents the human shape detection and content superimposition module, part of the Mobile Image Recognition based Augmented Reality (MIRAR) framework [10], one of the Mobile Five Senses Augmented Reality project's modules [12]. The development of an Augmented Reality (AR) system for museums, that acts as a guide for cultural, historical and museum's events, is the aim of the M5SAR project. Nowadays, almost every known museum has a mobile application (App), with or without AR systems, see e.g. [11]. The novelty of the M5SAR project is to extend the AR to the human five senses.

The development of a mobile multi-platform AR framework is one of the MIRAR module focus. The framework's goals are the following: a) to detect museum's pieces (e.g., paintings and statues) [10], b) recognize environments [16], and c) detect human shapes in order to overlay AR contents – Human Shape Superimposition sub-module. The detection of human shapes and the overlay of different clothes over those shapes is this paper's focus.

Recent years have witnessed significant progress in the detection of human shapes due to the use of Convolutional Neural Networks (CNNs). Two popular CNN frameworks for human shape detection and segmentation have stood out, namely the OpenPose [3] and the Mask R-CNN [7]. These frameworks represent the basis for structures more suited for mobile devices [5, 9, 13].

The OpenPose model is used to achieve the overlapping of different types of clothes on persons that are in real environments using a mobile device, being this the main contribution of this paper.

A initial algorithm for mobile human shape superimposition is presented. It is important to stress that, due to M5SAR project's restrictions, all software has to be implemented using Unity 3D [15], being OpenCV [4] asset for Unity used in the implementation. In the following sections, the development details for the human shape detection is explained in Section 2 and the superimposition process is detailed in Section 3. For last, conclusions and future work are drawn in Section 4.

2 Human Shape Detection

As mentioned above, one step of the MIRAR sub-module is the detection of human shapes. This implementation has to be accomplished in real-time on a mobile device, while the user is moving freely, which increases the computational complexity level [2]. An OpenPose model [9, 13] is used for detection, implemented on TensorFlow [6] and trained on the COCO dataset. In our case, the base CNN architecture for feature extraction is MobileNets [8]. The extracted features serve as input for the OpenPose algorithm, that produces confidence maps (or heatmaps) and part affinity fields (PAFs) maps which are concatenated. For COCO dataset, the concatenation consists of 57 parts: 18 keypoint confidence maps plus 1 background and 19×2 part affinity maps. The heatmaps represent the pixels confidence about a certain body part and the PAFs represent the limbs information in the x and y directions (2D vectors). Here, a component (joint) of the body like a right knee, the right hip, or the left shoulder (see Fig. 1, the red and blue circles, where blue indicates the person's right body parts) is a body part. A pair of connected parts, like the right shoulder connection with the neck (see Fig. 1, the green line segments) is a limb.

To evaluate the CNN two computational devices were used, namely an ASUS Zenpad 3S 10 tablet and a windows machine with an Intel i7-6700 CPU @ 3.40GHz. A total amount of 86 frames of expected user navigation were the input to the CNN. Furthermore, two input sizes images for the CNN were tested: 368×368 and 184×184 pixels (px). Depending on the size of the input, the average process time for each frame was 236 ms (milliseconds) and 70 ms, respectively, in the desktop, while in the tablet, the average process time for each frame were 2031 ms and 599 ms, respectively.

As expected, reducing the input images size of the CNN allows attaining improvements on the execution time, but the accuracy of the results dropped. One example of missing body part for 184×184 px than compared with 368×368 px is shown in [1]. Another problem noticed when using the two input sizes images is that, sometimes a confusion occurs between right and left hands/legs [1].

To solve these errors (confusion between right and left hands/legs and missing body parts) we expect to use "historical data" from previous pose estimations. For example, if in the previous five frames pose estimations are considered correct, then for each body part a "median" (estimation) for the x and y positions is done. Around this, is created a Region of Interest (RoI) that is used to decide if the next estimated body part(s) is/are right or not. Otherwise, it is used the "median" to substitute the wrong estimations. To solve the missing parts like wrists and elbows, it is expected to use the same procedure. Nevertheless, as these are parts moving more freely, it is also expected the need of additional work.

3 Clothes Overlapping

In this section, the method used to overlap clothes with textures is explained. The main steps of the algorithm are the following: (a) add skeleton to the textures, (b) resize textures, and (c) project textures over the person.

The first step, (a), is done to allow deformations on textures, by adding a skeleton to them. For this purpose, a 2D Skeletal Animation tool [14] is used to do a set-up of *bones* and automatically calculate the geometry and weights related to textures. The geometry is the number of vertices

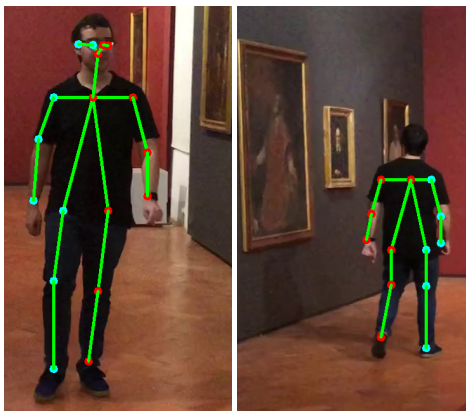


Figure 1: Examples of pose estimation.

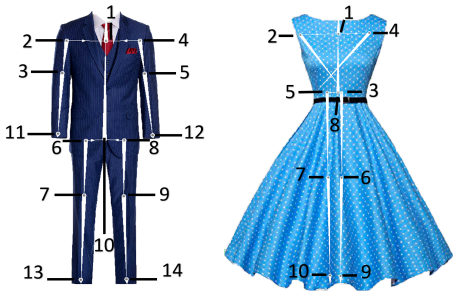


Figure 2: Example of created *bones*.

Figure 3: Example of human shape superimpostion using “textures".

attached to each bone. For example, if a bone moves the attached vertices do the same. A weight specifies of how much influence a bone has over a vertex. Setup of skeleton *bones* for a suit and dress are shown in Fig. 2. The number of *bones* defined for the suit is 14 and for the dress is 10. Additionally, the most common and correct outputs of body joints from OpenPose define the position of the *bones*. In the second step, (b), the distance between ankles and neck (an approximation to the person's height) is taken into consideration to resize the textures. Regarding step (c), the estimated body joints from OpenPose are used to place the textures. To rotate the texture bones, the angle (α) of each limp relative to a vertical alignment is calculated. In other words, the textures deformed over the person's body is the process achieved by placing each bone from texture skeleton over the respective body parts. The results showing two people overlapped with a dress and a suit are shown in Fig. 3.

To overlap textures over a person takes an average processing time of 29.31 ms on the mobile device and 6 ms using the desktop. In general, the complete process takes 559 ms plus 29.31 ms equals 588.31 ms in the mobile device and 70 ms plus 6 ms equals 76 ms in the desktop.

4 Conclusions

This work presents the initial results and proof-of-concept for a procedure to do human shape superimposition using textures. For future work, 3D clothes models will be used instead of 2D textures and more models for pose estimation will be tested.

Acknowledgments

This work was supported by the projects: FCT LARSyS UID/EEA/50009/2013, and M5SAR I&DT nr. 3322 financed by CRESC ALGARVE2020, PORTUGAL2020 and FEDER. We also thank Faro Municipal Museum and the M5SAR project leader, SPIC - Creative Solutions [www.spic.pt].

References

[1] R. Bajireanu, J.A.R. Pereira, R.J.M. Veiga, J.D.P. Sardo, P.J.S. Cardoso, R. Lam, and J.M.F. Rodrigues. Mobile human shape superimposition: an initial approach using openpose. *18th International Conference on Applied Computer Science, Dubrovnik, Croatia, 26-28 Sep.*, 2018. (accepted for publication).

[2] C. Bhole and C. Pal. Automated person segmentation in videos. In *Pattern Recognition (ICPR), 21st International Conference*, pages 3672–3675. IEEE, 2012.

[3] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[4] Enox. OpenCV for Unity. `https://goo.gl/MuxFj2`, 2018. Retrieved: August 28, 2018.

[5] Amit J. et. all. Enabling full body AR with Mask R-CNN2Go. `https://bit.ly/2jfnn8S`, 2018. Retrieved: August 10, 2018.

[6] Google. TensorFlow. `https://www.tensorflow.org/`, 2018. Retrived: January 14, 2018.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Computer Vision (ICCV), 2017 IEEE International Conference*, pages 2980–2988. IEEE, 2017.

[8] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] I. Kim. tensorflow-pose-estimation. `https://bit.ly/2HJxxcq`, 2018. Retrieved: August 10, 2018.

[10] J.A.R. Pereira, R.J.M. Veiga, M.A.G. Freitas, J.D.P. Sardo, P.J.S. Cardoso, and J.M.F. Rodrigues. MIRAR: Mobile image recognition based augmented reality framework. In *International Congress on Engineering and Sustainability in the XXI Century*, pages 321–337. Springer, 2017.

[11] Qualcomm. Invisible museum. `https://goo.gl/aS0NKh`, 2018. Retrieved: August 04, 2018.

[12] J.M.F. Rodrigues, J.A.R. Pereira, J.D.P. Sardo, M.A.G. de Freitas, P.J.S. Cardoso, M. Gomes, and P. Bica. Adaptive card design UI implementation for an augmented reality museum application. In *International Conference on Universal Access in Human-Computer Interaction*, pages 433–443. Springer, 2017.

[13] A. Solano. Human pose estimation using openpose with tensorflow. `https://goo.gl/7t7SGS`, 2018. Retrieved: August. 10, 2018.

[14] Unity. 2D Animation. https://bit.ly/2QjFwkn, 2018. Retrieved: August 28, 2018.

[15] Unity. Unity3D. `https://unity3d.com/pt`, 2018. Retrieved: August 27, 2018.

[16] R.J.M. Veiga, R. Bajireanu, J.A.R. Pereira, J.D.P. Sardo, P.J.S. Cardoso, and J.M.F. Rodrigues. Indoor environment and human shape detection for augmented reality: an initial study. *In Procs23rd edition of the Portuguese Conference on Pattern Recognition, Aveiro, Portugal, 28 Oct., pp. 21.*, 2017.

Paper:#3

QBER Compensation due to Polarization Drift using Quantum Machine Learning

By: Gonçalves, C. Belo, D. Almeida, L. Ramos, M. Jordão, M. Georgieva, P.

QBER Compensation due to Polarization Drift using Quantum Machine Learning

Cristiano Gonçalves
cristianogoncalves@ua.pt
Daniel Belo
dgb@av.it.pt
Luis Almeida
luisfilipealmeida@ua.pt
Mariana Ramos
marianaferreiraramos@ua.pt
Marina Jordão
marinajordao@ua.pt
Petia Georgieva
petia@ua.pt

University of Aveiro, Department of Electronics
Telecommunications and Informatics
Aveiro, Portugal

Abstract

A major problem of polarization coding in quantum communication systems is the polarization dispersion due to the birefringence effects in the optical fiber link. In order to keep a correlation between the State of Polarization (SOP) at the input and output of a transmission link, some type of polarization stabilization is needed. In this paper we propose a machine learning (ML) approach to compensate the polarization drift and recover the quantum states at the receiver. The problem is formulated as an inverse classification task, where the SOP at the receiver is used to recover the SOP of the encoded photons at the transmitter. Support Vector Machine (SVM) models using Polynomial and Gaussian Kernels are demonstrated to be efficient alternatives that can avoid additional hardware often applied as active polarization stabilizers. These results represent a step towards ML-based applications in quantum information processing.

1 Introduction

The quantum information problem that inspired this paper is the quantum cryptography that is researched as a solution to provide both privacy and security of internet data [1]. In this framework the information is transmitted using single-photons modulated in two-orthogonal states of polarization, [2]. However, due to the birefringence effects in the optical fiber link (i.e. delay on polarization states as the light propagates), polarization deviation occurs and increases the Quantum Bit Error Rate (QBER). In order to keep a correlation between the State of Polarization (SOP) at the input and output of the transmission link, an additional hardware is often implemented to compensate the drift in polarization over the optical communication link, [3].
Recently, ML techniques have been applied to compensate the fluctuations of the physical parameters of the signal in quantum key distribution systems with continuous variables, such as intensity and phase of the laser [4]. In this work we propose for the first time a Support Vector Machine (SVM) approach to decode the SOP of the emitted photon from the noisy measurements at the receiver. SVM holds the promise to become an efficient alternative, to the hardware solutions, as a polarization drift equalizer and therefore decrease the QBER over the quantum communication protocols.

2 Data transmission setup

In the data transmission setup the information is propagated using binary encoding of a single-photon modulated in four possible linear polarization states: horizontal, vertical, diagonal (45°) and anti-diagonal (−45°). A convenient way to visualize the polarization states is the Poincare sphere (Fig.1). Quantitatively, the polarization modes are expressed by the Stokes Parameters where S_0 corresponds to the beam intensity, represented as the radius of the Poincare sphere. The other Stokes parameters (S_1, S_2, S_3) define the polarization directions. The normalized Stokes vector (corresponding to the unitary power of the beam) is obtained after division of S_1, S_2 and S_3 by S_0. Thus, the tree normalized components of the Stokes vector are the coordinates of the polarization states in the 3D space. The transmission system built to acquire labeled data consists of a transmitter

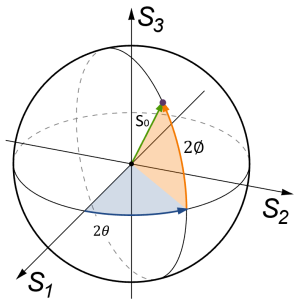


Figure 1: Poincare sphere

that sends encoded photons using one of the polarization states. The encoded photon is propagated through the optical fiber link and due to the birefringence effects caused by optical fiber conditions it suffers a deviation in its SOP. The photon obtained at the receiver has a different SOP.

Fig. 2 shows the transmitted data (Noise free data in red circles) and the received data, randomly divided into training (blue dots) and test (red dots) data during the data modeling. The simulated angular noise in the optical link causes significant data spread at the receiver and as a result a high QBER of the overall system. Four possible states of polarization

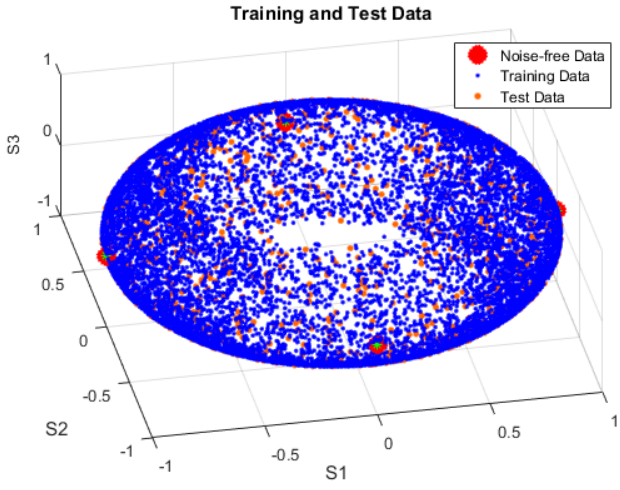


Figure 2: 3D visualization of training and test data.

were transmitted, with 3000 symbols per each state, or 12000 samples in total. Data were randomly divided into 80% for training and 20% for testing.
The compensation of the polarization drift through the optical fiber is formulated as a classification problem of four classes (the four possible photon polarization states at the transmitter) based on three inputs (features) - S'_1, S'_2, S'_3 and given labels $Y_{label} = (S_1, S_2, S_3)$. Provided with samples measured at the receiver (the received photon positions) the SVM should output the correct Y_{label}.

3 Results

In this work, multiclass SVM with Linear, Polynomial and Gaussian kernels are comparatively studied. SVM was chosen due to its performance, intuitive interpretation and flexibility to fit to problems from different fields [5]. Its major characteristics is the maximization of the margin between the two classes.

3.1 SVM with Linear Kernel

Linear kernel SVM (dot product of input space vectors x_i and x_j):

$$K(x_i, x_j) = (x_i)^T (x_j) \tag{1}$$

The results are depicted in Fig. 3. As can be seen due to the linear kernel the volume attributed to each class is a 3D rectangular shape.

3.2 SVM with Polynomial Kernel

Polynomial kernel SVM:

$$K(x_i, x_j) = (x_i^T x_j + c)^p \tag{2}$$

where c ≥ 0 is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial. The results of decoding the photon polarization states are visualized in Fig. 4. Due to the higher order kernel, more complex shaped decision boundary is achieved. In this implementation 3rd order polynomial kernel revealed to be a good compromise between training data fitting and model generalization property. The volume attributed to each class fits better the experimental data than with the linear kernel. The few miss-classified points are the ones affected by a high noise amplitude.

3.3 SVM with Gaussian Kernel

Gaussian kernel SVM :

$$K(x_i, x_j) = exp(- \parallel x_i - x_j \parallel^2 / 2\sigma) \tag{3}$$

The classification results are shown in Fig. 5. They are similar to the polynomial Kernel SVM since the Gaussian function behaves as a high order polynomial. The classification accuracy on test data using different SVMs are summarized in Table 1.

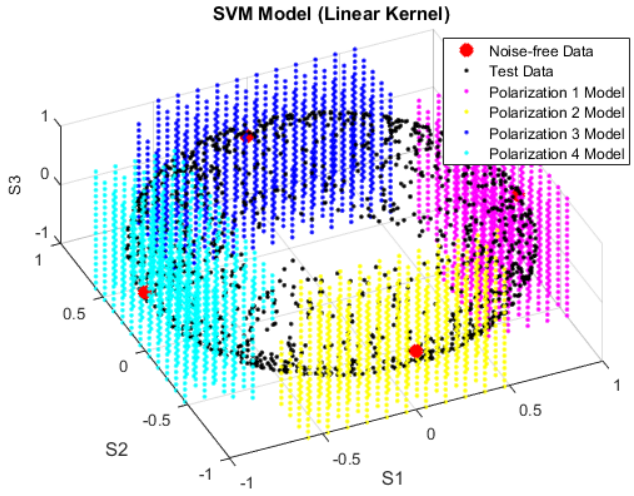


Figure 3: Polarization states decoding with Linear Kernel SVM.

Table 1: Performance results on test data

Method	Accuracy
SVM Linear Kernel	76.5%
SVM Polynomial Kernel	90.5%
SVM Gaussian Kernel	89.5%

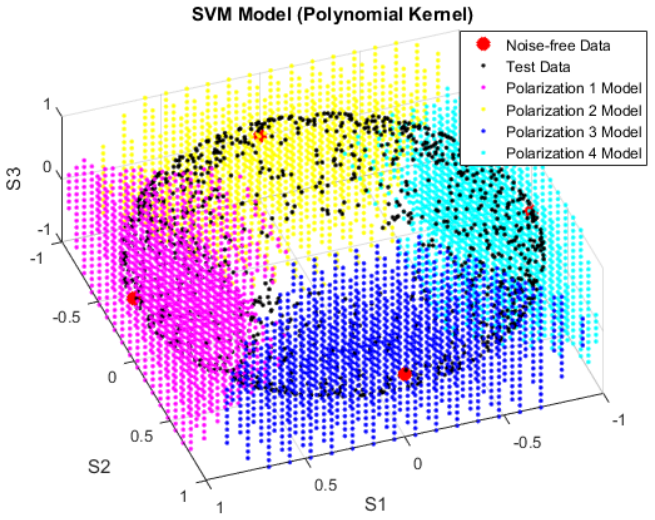


Figure 4: Polarization states decoding with Polynomial Kernel SVM.

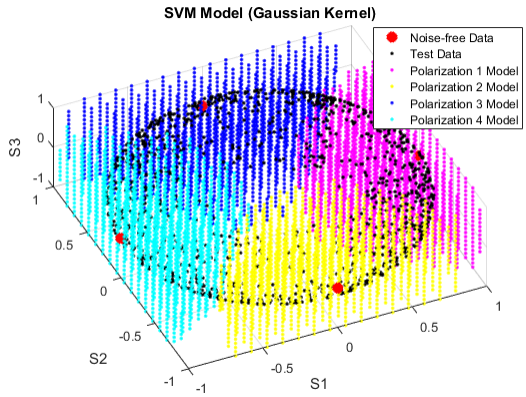


Figure 5: Polarization states decoding with Gaussian Kernel SVM.

4 Conclusion

This work was focused on studding the applicability of a ML approach for polarization drift compensation in quantum communication channels. It was shown that the choice of kernel is relevant for the performance of the SVM algorithm. Nonlinear kernels are favorable for this problem. Better results were obtained for SVM with Polynomial and Gaussian Kernels. The experiments demonstrated that polynomial Kernel SVMs achieved above 90 % accuracy in recovering the photon polarization states.

References

[1] Yao, Andrew Chi-Chih (1995). "Security of quantum protocols against coherent measurements". Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, ACM: 67–75.

[2] Yu He, Y.-M. He, Y. Wei, X. Jiang, K. Chen, C.-Y. Lu, J.-W. Pan (2017). Quantum State Transfer from a Single Photon to a Distant Quantum-Dot Electron Spin. Preprint arXiv:1706.08242.

[3] G. B. Xavier, N. Walenta, G. Vilela de Faria, G. P. TemporÃčo, N. Gisin, H. Zbinden, J. P. von der Weid (2009) Experimental polarization encoded quantum key distribution over optical fibres with real-time continuous birefringence compensation. New Journal of Physics -Quantum cryptography: Theory and Practice, vol.11, 2009.

[4] Liu, W., Huang, P., Peng, J.; Fan, J. and Zeng, G. (2018). "Integrating machine learning to achieve an automatic parameter prediction for practical continuous-variable quantum key distribution". Physical Review A, APS. 97 (2): 022316.

[5] Cortes, C., V., Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273 - 297.

Paper:#14

Learning Anticipation Skills for Robot Ball Catching

By: Carneiro, D. Silva, F. Georgieva, P.

Learning Anticipation Skills for Robot Ball Catching

Diogo Carneiro
diogo.carneiro@ua.pt
Filipe Silva
fmsilva@ua.pt
Petia Georgieva
petia@ua.pt

University of Aveiro, Department of Electronics
Telecommunications and Informatics
Aveiro, Portugal

Abstract

In this work we studied the impact of using the information provided by the thrower's motion for improving the success rate of a robotic system catching flying balls. The anticipation mechanism proposed is based on a neural network model that predicts early enough the initial position and velocity of the ball at the moment it is released.

1 Introduction

The interception of a flying object along its trajectory, either in humans or robots, is a challenging task due to demanding spatio-temporal constraints, requiring the coordination among visual, planning and control systems. The successful accomplishment of the task involves a sequence of control actions, including moving the hand to the interception point, adjusting the hand's orientation and closing the hand [1], [3], [2]. Given the short ball's flight time, it seems advantageous to make predictions as early as possible so that the robot planning stage has enough time to find a valid catching posture.

The goal of this work is to study the role of anticipatory mechanisms for robot ball catching. The idea is that an effective solution to the ball-catching problem should involve all relevant information as early as possible and not only information extracted during the flying phase. In this context, early anticipations refers to taking prior actions on the basis of information extracted during the period in which the human is preparing to throw the ball towards the robot catcher.

The paper presents a comparison between the results obtained using early predictions and those using the classical methodology relying solely upon the information extracted during the flight phase. The motion of the thrown object is divided into two phases: (i) the motion before it is released (preparatory motion) and (ii) the motion after the release (ballistic motion). Here, we propose a feedforward neural network (FNN) to estimate the release position and velocities of the projectile based on the intentions of the human partner during the preparatory motion. The subsequent predictions of the ball's trajectory enable the robot to react as soon as possible, improving the ball catching performance.

2 Dataset Generation

The testbed for the ball catching experiments is shown in Fig. 1. The reference frame is defined such that the coordinate system S_r is fixed to the ground with the z-axis oriented upwards and aligned with the first axis of the robot arm, the x-axis oriented towards the human thrower and the y-axis transversal to that directions. The robot arm is located at a height of 1.40 m above the ground. The human partner performs an underhand throw in the robot's direction, whereas the trajectory of the flying ball is modeled as a parabola (air resistance is neglected).

The system is divided into three consecutive steps. First, the ballistic motion is specified considering the following input parameters: (i) the desired initial position of the ball (i.e., the final position of the thrower's hand), (ii) the desired final coordinates of the ball when it intercepts the ground plane, and (iii) the desired ball's flight time. These input parameters allow computing the initial velocity of the ball that corresponds, given continuity conditions between preparatory and ballistic phases, to the final velocity of the thrower's hand.

The initial coordinates of the ball $P_i = (x_i, y_i, z_i)$, with respect to the reference frame and the final coordinates of the ball on the ground plane $P_f = (x_f, y_f, z_f)$ are uniformly distributed random numbers in the intervals: $x_i \in [3.5; 3.75]$ m, $y_i \in [-0.1; 0.1]$ m and $z_i \in [0.8; 1.4]$ m, $x_f \in [-3.0; 0.0]$ m, $y_f \in [-1.0; 1.0]$ m and $z_f = 0$ m. The flight time is also randomly generated (uniform distribution) in the interval [0.5; 1.2] s.

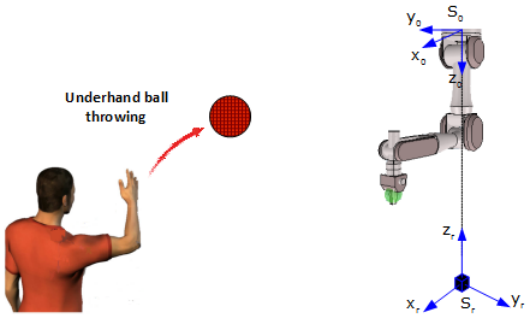


Figure 1: Testbed for the human-robot ball catching experiments.

Second, the generation of the thrower's hand motion is based on a demonstration extracted from real data. Human underhand throwing data was recorded from a VICON system with 8 infrared cameras. The 3D coordinates were collected at 100 Hz. Then, polynomial interpolation is used for generalizing the demonstration to new situations, considering the restrictions imposed by the initial position and velocity of the ballistic phase. Gaussian noise is added to the generated trajectories to account for the effects of noisy measurements.

3 Methodology

The anticipation mechanism considers two sequential motion phases, reflecting the ball trajectory before and after it has been released:

- **Phase A** - Opponent's hand throwing motion (preparatory motion). During this phase the robot learns to predict as early as possible the initial position and velocity of the ball at the moment it is released (the anticipation phase).
- **Phase B** - Free-flying ball motion (ballistic motion). During this phase the robot learns to predict the most favorable interception point in its reachable space.

The state of the ball in phase A is represented as $S^{(A)} = \{S^{(A)}_{pos,t}, S^{(A)}_{vel,t}\}_{t=1,...T}$, where $S^{(A)}_{pos,t} \in \mathbb{R}^3$ denotes the ball position and $S^{(A)}_{vel,t} \in \mathbb{R}^3$ denotes the ball velocity at the moment t. T is the terminal point of phase A, which is also the starting point of the free-flying ball phase.
Learning at the anticipation phase is designed as a function approximation between visually acquired information of the opponent's hand movements (the first L samples of the ball state $S^{(A)}$) and the ball position and velocity at the moment it is thrown in the air,

Two FNNs are trained to estimate independently the initial position, $\bar{S}^{(A)}_{pos,T} \in \mathbb{R}^3$, and the initial velocity, $\bar{S}^{(A)}_{vel,T} \in \mathbb{R}^3$, of the ball in-flight,

$$\bar{S}^{(A)}_{pos,T} = f_{FNN1}(S^{(A)}_{pos,t}, S^{(A)}_{vel,t}, t = 1, ...L, L << T) \quad (1)$$

$$\bar{S}^{(A)}_{vel,T} = f_{FNN2}(S^{(A)}_{pos,t}, S^{(A)}_{vel,t}, t = 1, ...L, L << T) \quad (2)$$

After observing the first L samples of the thrower motion, the predictive models (1) and (2) provide estimation of the initial position and velocity of the ball in-flight which triggers much earlier estimation of the flying ball trajectory and ultimately suggests a feasible ball catching point.

Latter on, when the ball enters phase B, the robot continues to adjust its catching position based on dynamically collected information for the

flying ball. However, if the anticipation phase is successful, the robot is better prepared for the incoming ball. The robot arm is already close to a feasible catching point, which improves significantly the interception success rate. N training trajectories with L data points are used to fit the regression models.

The final structure of the NN models, in terms of number of input samples L, number of hidden layers, number of layer nodes, were defined as k-fold Cross Validation (CV) optimization on data . 1500 demonstrations of the ball catching task were generated with a sampling frequency of 100Hz. In order to account for the inherent measurement noise, a sensor with 40dB SNR was used in the simulations. The FNN models were trained with 1000 throws (randomly divided into 75% for training and 25% for validation). The performance measure is the batch mean squared error (MSE) between the multidimensional ($\mathbb{R}^3$) target and the model predictions.

4 Simulation experiments

This section evaluates the effect of various parameters on the catching success rate, including the noise level in sensory feedback and the maximum velocities available at the robot's joints. The study compares the catching performance with and without early predictions.

4.1 Perception-Action Trade-off

Despite the role of the preparatory phase for anticipating the robot's action, the ballistic phase is important to refine the prediction of the catching point as more data is available. The question is whether waiting (switching time) for a confident prediction is advantageous or, instead, it may cause delayed reactions that prevent the robot from reaching its target. Fig. 2 shows the effect of the robot's decision when to start moving on the catching performance. The switching time is expressed as the number of observations (samples) of the ball in-flight. The success rate is evaluated with the maximum joint velocity limited to 135 deg/s and 40dB SNR in the sensory signals.

For the classical strategy (red line), the highest success rate occurs when the switching time is in the first 10-samples of the flying ball. Presumably, later the robot arm starts moving, less likely is to catch the ball. The anticipation strategy (blue line) shows a similar behavior i.e. there is no advantage in re-planning the robot's motion before the 18th-sample. Note that the anticipation method outperforms the classical one in about 10% over the full range of values.

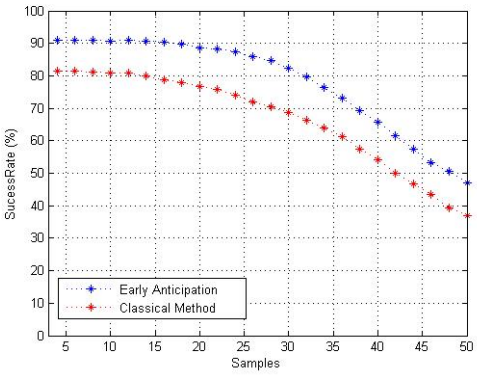


Figure 2: Success rate as a function of the switching time for the early prediction (blue) and the ballistic prediction (red) strategies.

4.2 Robustness Against Noise in Sensory Signals

Fig. 3 compares the success rate as a function of the SNR level on the sensory data for a switching time of 20-samples and a maximum joint velocity of 135 deg/s. First, early predictions outperform up to 20% the classical methodology solely based on ballistic predictions. Second, these results demonstrate the increased robustness against noise provided by the FNN: in the range of 36 dB to 50 dB the performance remains practically unchanged (the mean value is 93.4 and the standard deviation is 1.47). In contrast, the classical method is more sensitive to noise with a significant impact on the performance for SNR below 42 dB.

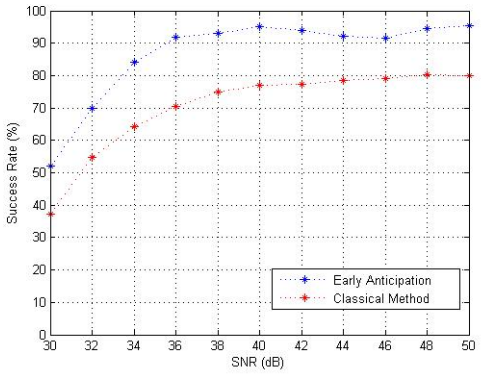


Figure 3: Success rate as a function of the SNR for the early prediction (blue) and the ballistic prediction (red) strategies.

4.3 Impact of the Maximum Joint Velocities

Fig 4 shows the success rate as a function of the maximum velocity, assuming the same bound for the three joints. 10 samples switching time and 40dB SNR were set up for the simulations.

As can be observed, the two methods show an approximately linear relationship between the task performance and the maximum angular velocity within the range of 45 to 135$^\circ/s$. The advantage of using early predictions is clearly noticed with improvements in the success rate which reach the 10-12% over almost the entire range.

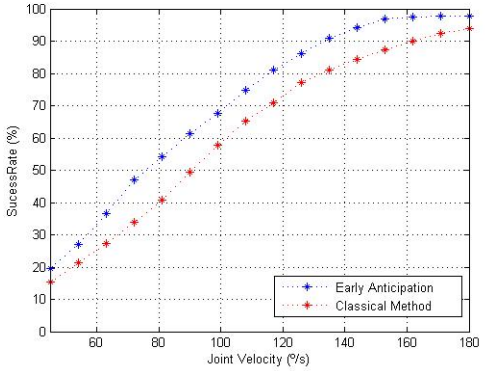


Figure 4: Success rate as a function of the maximum joint velocities for the early prediction (blue) and the ballistic prediction (red) strategies .

5 Conclusions

Intention inference from observation of human actions is an essential ability for robots performing interactive tasks. This work studies the role of early anticipation skills to improve the performance of a robotic system playing ball catching with a human partner. The source of anticipatory information results from the observation of the thrower's motion before the ball is released. For that purpose, a FNN is trained to estimate the initial position and velocity of the ball in-flight given a sequence of observations during the throwing phase. The proposed approach outperforms up to 20% the classical methodology in which the generation of predictions solely relies upon the available information during the flight phase.

References

[1] P. Cigliano, V. Lippiello, F. Ruggiero, and B. Siciliano. Robotic ball catching with an eye-in-hand single-camera system. *IEEE Transactions on Control Systems Technology*, 23(5):1657–1671, 2015.

[2] S. Kim, A. Shukla, and A. Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.

[3] S. Salehian, M. Khoramshahi, and A. Billard. A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Transactions on Robotics*, 32(2):462–471, 2016.

Paper:#18

Sensor-based Activity Recognition on Smartphones: A Simple Approach for Sharing Results with Other Applications

By: Andrade, R. Gonçalves, P. Alves, A.

Sensor-based Activity Recognition on Smartphones: A Simple Approach for Sharing Results with Other Applications

Renato Andrade
renatoandrade@dei.uc.pt
Center for Informatics and Systems of the University of Coimbra, University of Coimbra

Paulo Gonçalves
pafgoncalves@ipc.pt
Informatics and Systems Engineering Department, Coimbra Institute of Engineering

Ana Alves
ana@dei.uc.pt
Center for Informatics and Systems of the University of Coimbra, University of Coimbra

Abstract

This work has as main objective, to employ the most recent techniques used for activity recognition based on sensors. In the scope of this study, the work designed was based on common sensors in mobile devices, having been developed two applications for the Android operating system. The main one is able to recognize the activities and send this information to the system, allowing it to be used by any application installed in the device that is prepared for such purpose, while the other is only a proof of concept and aims to receive the information sent by the first one and display it to the user.

1 Introduction

Human Activity Recognition (HAR) is a research field that seeks to identify actions performed by individuals through the context and the environment in which they are[1]. It has seen a very large development in recent years, mainly due to the ease of access to various types of sensors present in mobile devices and to their large and increasing processing capacity[2]. Such sensors, commonly found in smartphones for example, are widely used as the main means for the recognition of activities in scientific studies and in software already developed and available in the market[3].

Initially, the work carried out in this field, with the objective of developing applications capable of recognizing activities, typically involved the need for an externally trained model, which was then introduced into the application and started to be used to recognize the activities. Afterwards, studies began to emerge that proposed not only the recognition of the activities on mobile devices, but also the data collection and the training of the model carried out online, that is, internally in the application. Thus the need for the data to be manipulated outside the devices is eliminated, which solves issues related with data privacy and results in "self-contained" software, being also more efficient [4].

There are several studies on the subject, namely [1], [2], [4], [5], [6] and [7], where the whole process of data collection, training and classification is done on the device without having to send data to an external server to do the processing.

In [1], [2], [5], [6] and [7], the list of activities to be recognized is predetermined and there is no possibility of including new activities. The authors of [4] add this capability, allowing the user of the application to select a new activity, collect data, train and detect it. Even so, other system applications cannot take advantage of the first application's ability to detect activities. The Android system has an API [8] that allows the detection of user activity and the use of this recognition by other applications in the system but is limited to a relatively small and predetermined list of activities.

The main application developed in this work, allows the user: to choose the activities that he want to train, to collect data and train a classification model when the user performs each one of these activities, to put the application in detection/test mode and, finally, and different from other authors, to trigger an "event" upon detection of an activity so that other applications present in the system can obtain this information. This "listener" application may use this information for any purpose. It is also possible, if the user chooses, to register a new activity that does not exist in the application or possibly exclude activities that no longer intends to use.

2 Application "PatternRecognition"

It was developed a system in which each user could do their own activity recognition training. This allows better adaptation to the characteristics of the way the data is collected, such as the position of the smartphone. In addition, a more personalized training avoids problems that eventually happen in recognition, caused by how each user performs a particular activity, since distinct users perform activities differently. Another advantage is the possibility of repeating the training if the classifier is not giving results with the desired precision.

Another goal was to develop a system that would allow other applications in the mobile device to take advantage of activity detection. To this end, when the application detects an activity, in addition to displaying it to the user, it also launches an event for the Android operating system (a broadcast[9] message) that other applications can use (implementing a "Broadcast Receiver"[10]). In this way, the user can install this application, train it to recognize any activity that can be classified through the movement of the smartphone, which is totally independent of the applications that depend on this functionality. This allows the lifecycle of applications that use this functionality to be completely different, just by having the application "PatternRecognition"[11] installed so that they can use it, work without relying any additional computation at compile time.

The user can choose the activities that he wants to detect, collect the data for training while performing the same activity, then train the application with this data and classify the activities he is doing in real time, as shown in Figure 1.

The application developed consists of 5 functionalities:

- **Data Collect**: where it is possible to gather data properly labelled for training.
- **Train**: where tagged data previously gathered is used to create 4 models of which the application itself chooses the one that has the best performance for future classification.
- **Activity Classification**: where, on a constant basis, data collected are classified as follows.
- **Activity Management**: where you can add and remove activities to be recognized.
- **Sensor Listing**: where it is possible to list the sensors of the current mobile device.

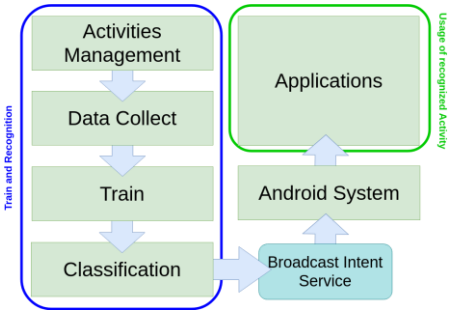


Figure 1: Stages involved in the operation of the application, from the moment of its installation, to the recognition of activities and broadcast messages for the other applications of the device.

In the scope of this work, a second application, called "BroadcastReceiver"[12], was also developed. This application is basically a proof of concept, whose objective is to demonstrate how it is possible to receive the information sent by the application "PatternRecognition".

2.1 Architecture

The application consists of a series of Android Activities (Windows) related to each of the main stage mentioned above. There is a service, named “Sensors Service” as presented in Figure 2, that receives data from sensors and transmits them to be pre-processed before being sent to a classifier (when the application is in activity detection mode) or to another class that stores the data for later training (when in data collection mode). If the system is in classification mode, after recognition of the activity, the system broadcasts an "Intent"[13] with the acknowledged activity information. That way any other application in the system can use this information.

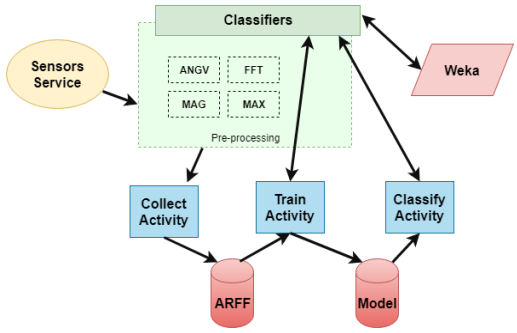


Figure 2: Architecture of the "PatternRecognition" application.

2.2 Classification

In the scope of the study, the main focus was therefore the functionality of the recognition of activities and not the method as such recognition is made, since there are already several libraries and a wide range of algorithms and studies of these algorithms, which demonstrate the efficiency, the advantages and the disadvantages of each of them comparing themselves. Therefore, in an earlier point of this study, which will not be addressed in this work, comparisons were made of the results of several classifiers, as well as their performance and execution time, to determine the best 4 algorithms to be used in this work. Thus, classification and creation of the respective classification model is done using the Weka library [14], a collection of machine learning algorithms for data mining tasks. The data gathered in the collection phase is stored in a ARFF file[14], which is then used to train the "J48", "RandomForest", "PART" and "REPTree" algorithms. These algorithms were the ones that showed the best relation between performance and execution time. During training, the application chooses as the classifier to be used, the algorithm that presents the best results.

As previously mentioned, after data collection, training and generation of a classification model based on these data are carried out, so that the application can perform activity recognition. However, both data collection and classification involve the application of some pre-processing of data to allow greater accuracy in detection. Thus, at the time of data collection, the data is pre-processed, then stored in the ARFF file and, the data go through the same pre-processing before being classified.

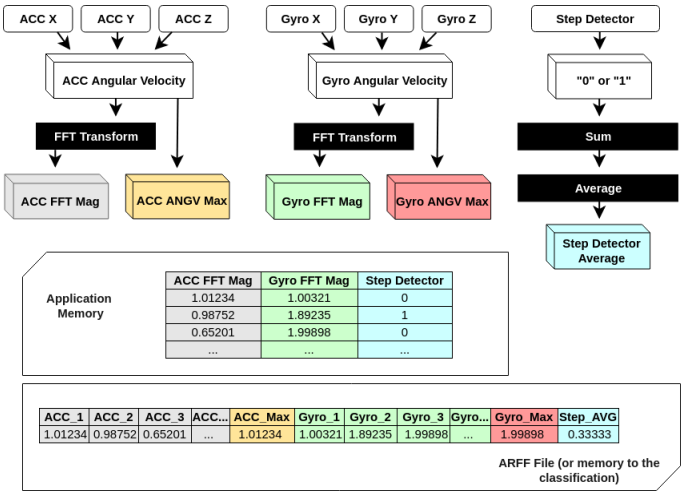


Figure 3: Pre-processing made to the data at the time of collection and in the "activity detection" mode.

As shown in Figure 3, based on the sensor data, the following features are created: Angular Velocity, Fast Fourier Transform - FFT[15], Magnitude[15] and Maximum (the highest value presented in a given interval). However, to talk about features, it is first necessary to explain the data and the sensors used. The application uses the accelerometer, the gyroscope and the step detector. In the case of the first and the second, data is collected from the three axes that both have. Regarding the step detector, the application registers this sensor, the value "1" when a step is detected or "0" when it is not detected. This way, for each time one of these sensors changes, the application collects the current values from all of them and then calculates the angular velocity for the axes of the accelerometer and the gyroscope, keeping the results in memory variables, as well as the step detector data at that time. These values are stored in memory for 64 collection intervals, which are then transformed into a single row with multiple attributes. For this, the FFT is performed over the result of the calculation of the angular velocity of the values of the accelerometer and the gyroscope. Each result obtained in the Fourier Transform, then passes by the calculation of the magnitude. Each of the 64 lines obtained in the result of this calculation will then be transformed into a feature. The other features used are the maximum value recorded in these 64 samples and the mean of values recorded by the step detector.

3 Conclusions

Recent scientific advances have led to the emergence of several studies that have introduced in the market and in the academic area, various relevant techniques to the recognition of patterns and activities through data provided by different types of sensors found in almost all types of smartphones that currently exist. This study demonstrated in a simple and objective way, the application of some of these techniques in the development of applications for the Android operating system, adding an innovative concept that allows other applications in the device to benefit from the recognition of activities. In addition, the application presented here, allows the user to build and train the recognition of their own activity list, providing a more efficient result and more adapted to the usage patterns of their own mobile device.

4 References

[1] Ghio, A., et al. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. *J. UCS* **2013**, 19, 1295-1314.

[2] Ouchi, K.; Doi, M. Indoor-outdoor Activity Recognition by a Smartphone. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 2012; pp. 600-601.

[3] Shoaib, M., et al. A Survey of Online Activity Recognition Using Mobile Phones. *Sensors* **2015**, 15, 2059-2085.

[4] Frank, J.; Mannor, S.; Precup, D. Activity Recognition with Mobile Phones. *Lect. Notes Comput. Sci*. **2011**, 6913, 630-633.

[5] Stewart, V., et al. Practical automated activity recognition using standard smartphones. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops, Los Alamitos, CA, USA, 19-23 March 2012; pp. 229-234.

[6] Kose, Mustafa.; Incel, OD; Ersoy, C. Online Human Activity Recognition on Smart Phones. In Proc. of the Wksh. on Mobile Sensing: From Smartphones and Wearables to Big Data, 2012; 11-15

[7] Gomes, J., et al. MARS: A Personalized Mobile Activity Recognition System. In Proc. of the 2012 IEEE 13th International Conference on Mobile Data Management (MDM), 23-26 2012; pp. 316-319.

[8] Android Developers: Adapt your app by understanding what users are doing. Accessed September 3, 2018 at https://goo.gl/DFJ1Kp

[9] Android Developers: Broadcasts Overview. Accessed September 3, 2018 at https://goo.gl/P5P5XU

[10] Android Developers: BroadcastReceiver. Accessed September 3, 2018 at https://goo.gl/vR7CKb

[11] GitHub: PatternRecognition. Accessed September 3, 2018 at https://github.com/RibeiroSt/PatternRecognition

[12] GitHub: BroadcastReceiver. Accessed September 3, 2018 at https://github.com/RibeiroSt/BroadcastDetector

[13] Android Developers: Intent. Accessed September 3, 2018 at https://developer.android.com/reference/android/content/Intent

[14] The University of Waikato: Weka 3: Data Mining Software Java. Accessed September 3, 2018 at https://www.cs.waikato.ac.nz/~ml/weka/

[15] Shin, I., et al. A Novel Short-Time Fourier Transform-Based Fall Detection Algorithm Using 3-Axis Accelerations. *Mathematical Problems in Engineering*, 2015, 394340, 1-7

The authors would like to thank the funding by URBY.SENSE project (POCI-01-0145-FEDER-016848). URBY.SENSE is co-financed by COMPETE 2020, Portugal 2020, FEDE)and Fundação para a Ciência e a Tecnologia (FCT).

Paper:#22

KCentres algorithm in GPU for clustering on many-core architectures: A preliminary approach

By: Uribe-Hurtado, A. Orozco-Alzate, M. Lopes, N. Ribeiro, B.

KCentres algorithm in GPU for clustering on many-core architectures: A preliminary approach

Ana-Lorena Uribe-Hurtado
alhurtadou@unal.edu.co
Universidad Nacional de Colombia – Sede Manizales
Manizales, Colombia

Mauricio Orozco-Alzate
morozcoa@unal.edu.co
Universidad Nacional de Colombia – Sede Manizales
Manizales, Colombia

Noel Lopes
noel@ipg.pt
UDI, Polytechnic of Guarda, Portugal
CISUC – University of Coimbra, Portugal

Bernardete Ribeiro
bribeiro@dei.uc.pt
CISUC – Department of Informatics Engineering
University of Coimbra, Portugal

Abstract

Clustering is one of the most common tasks in pattern recognition. In many applications, such as those of the current Big Data era, it is vital that the clustering process is made as fast and efficient as possible. In this paper we present a first attempt for the parallel implementation of the KCentres clustering algorithm, based on a many-core (GPU) architecture and aimed to speed up the execution of the algorithm in comparison with a CPU-based version. The GPU-based algorithm was tested on a collection of data sets for shape clustering. Results are particularly enhanced for data sets having more than 1000 objects to be grouped.

1 Introduction

Hardware has undergone a faster evolution than the one experienced by software; as a result, heterogeneous architectures are easily available nowadays in personal desktop computers. In them, it is increasingly common to find built-in graphic processing units (GPUs) which, even though originally meant to speed up the rendering of the image frames in video games, have the added potential of executing instructions for multiple purposes: a type of computation that is currently known as GP-GPU (General-Purpose GPU). In addition, the combination of different architectures such as those based on either GPU or CPU, allows us to enhance the behavior of many still in-use and sequentially-designed computer programs.

In the so-called Big Data era [4], speeding up computer programs in general and those used for data processing in particular, is a major challenge currently faced by computer programmers worldwide. Among them, clustering algorithms are particularly relevant since they perform one of the most important data processing tasks. In this field, adapting sequential source codes to multi-core (CPU) and many-core (GPU) architectures provides the means to make the best of the hardware available and, therefore, to deliver results to the users in a much faster way, allowing researchers to tackle larger datasets.

In this paper we present a preliminary approach to the implementation of the KCentres clustering algorithm on a small-sized GPU: a Quadro M2000. For the sake of illustration, we show results for a number of problems corresponding to the task of grouping objects according to their shapes. The implementation strategy of the GPU-based algorithm, as well as experimental results for six publicly-available data sets, are shown.

The structure of the paper is as follows. Section 2.1 describes the sequential algorithm. Its corresponding GPU parallelization strategy is explained in Sec. 2.2. Section 3 describes the architecture used for the execution of the experiments as well as the experimental results. Finally, our conclusions are shown in Sec. 4.

2 KCentres clustering algorithm

KCentres can be considered as a variant of the well-known Kmeans algorithm [3], [6]. It was later popularized in [1] for the task of prototype selection in dissimilarity-based classification and as a conventional clustering method. In contrast to Kmeans, KCentres is entirely based on dissimilarities since it finds the centres just in function of the radius or distance from each object to the others (a companion feature representation is not required). In brief, the main idea of KCentres consists in designating as centres those objects whose maximum distance to all the other objects is minimum.

2.1 KCentres in CPU

The CPU-based sequential implementation is carried out as follows. It is assumed that a square matrix $D(N,N)$, containing all the pairwise dissimilarities between the N objects of the data set, has been pre-computed:

1. A set of K objects, out of N ones, is selected at random. They constitute the initial centres and are stored in a vector p. The indexes of entries in p are $j = \{0,\ldots,K-1\}$.
2. For all the N objects in the data set, the nearest centres are found and the indexes of the corresponding associations are stored in a vector of labels l. The indexes of l are $i = \{0,\ldots,N-1\}$.
3. Afterwards, in order to update the centers which are now stored in a vector c, for each group (objects having the same labels l_i) the new centre c_j is found as the object whose maximum distance to all the objects in the current group is minimum.
4. The algorithm converges in case that the updated centers c are the same previous ones (p); otherwise, it returns to Step 2 storing in p the centers found in Step 3.

2.2 KCentres in GPU

Three kernels —functions that are executed on the GPU— were created in order to implement the algorithm on a many-core platform. The implementation strategy consists in finding storage structures allowing computations with the GPU threads and, at the same time, guaranteeing that there are no dependencies among the data to be modified. Each kernel uses implicit synchronization and performs the following computations:

1. A first kernel, called `EDMGPU`, computes the distance matrix $D(N_w,N_w)$ on the GPU. This matrix is computed from the original feature representation of the objects. Without loss of generality, we have restricted ourselves to the Euclidean distance. Its computational complexity, using threads, reduces to $\mathcal{O}(N)$.
2. `findfirstCentres` —A function in CPU— randomly assigns the first centers without applying any parallelization. The computational complexity in this step is $\mathcal{O}(N)$ due to a permutation process over the N indexes of the objects required to find K unique centers. This function is carried out in CPU, using heterogeneous computation, because its implementation on GPU is not profitable for the value of K, which is typically small.
3. The second kernel, called `KCentresGPU`, is used for finding the nearest neighbor (Step 1 in the sequential implementation) as well as the maxima which are stored in a matrix M. (the radii, which correspond to Step 2 in the sequential implementation). We used N GPU threads, where $i = \{0\ldots N-1\}$ are the indexes of the threads. In order to perform this computation, each thread i finds each label from l and stores the results in l_i. Afterwards, each thread i computes the maximum distances for each one of the N objects in function of the repeated labels in l and, then, they are stored as an entry of the matrix M, i.e.: in $M_{k,i}$. Since each thread iterates over l in order to find coincident labels, this kernel has a computational complexity of $\mathcal{O}(N)$.

   The third kernel, `findminimumofmaximainGPU`, is used for computing the new centers (Step 3 in the sequential implementation). This kernel uses K threads from the GPU. Each thread, k,

looks for the minimum value in the matrix of maxima M, whose size is $K \times N$. Each thread updates its own new center c_k independently from the computation of the other ones. the computational complexity in this case is $\mathcal{O}(N)$.

4. Finally, a verification for convergence is carried out. It consists in checking whether the new centers are the same ones found in Step 1 of the implementation ($p_k == c_k$). This verification is carried out in CPU, i.e. using heterogeneous computation, due to the dependencies involved among the structures as well as because it is performed only once; thereby, it is not required that all the threads in GPU perform the same verification. In case that the verification produces a false boolean result, the data structures for l, M and c are restarted to prepare them for the subsequent iteration. This implies that they must be copied again to the GPU.

3 Experiments and Results

In order to test both implementations of the KCentres algorithm in GPU and CPU, a Hewllet Packard computer with 4 Intel (R) Xeon (R) E5-2603 v4 processors at 1.70GHz each was used, with a Quadro M2000 GPU with 1024 threads per block, 6 Multiprocessor Streaming (SM) and 32 CUDA cores.

Table 1 shows the main properties (number of centers to be found K, number of objects N and size of the distance matrix D: $N \times N$) of the six data sets of shapes used in the experiments[1]. The corresponding results obtained in the clustering process are presented in Table 2, namely: the number of iterations (I) performed by the algorithm in CPU and GPU until convergence —initializing both algorithms with the same set of initial centers p—, the elapsed times in seconds (ET) taken by each implementation and, finally, the speed up (S/P), reached with the execution in GPU. Experiments were executed five times per data set, such that the reported ETs are the averaged values across the executions.

Table 1: Data sets description

Data set	K	N	$N \times N$
Pathbased	3	300	90000
Jain	2	373	139129
Compound	6	399	159201
R15	15	600	360000
Aggregation	7	788	620944
D31	31	3100	9610000

Figure 1: Elapsed times

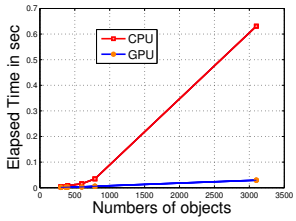


Table 2: Results of the KCentres algorithm in both CPU and GPU

Data set	CPU-GPU I	ET - CPU	ET - GPU	Speed Up S/P
Pathbased	6	0.0037	0.0015	2.41
Jain	3	0.0049	0.0020	2.49
Compound	4	0.0082	0.0023	3.61
R15	9	0.0153	0.0030	5.11
Aggregation	7	0.0346	0.0057	6.08
D31	11	0.6308	0.0293	21.51

Notice that the elapsed times of the implementation in the GPU are better (lower) than those in CPU, even for the case of the smallest data set. The best acceleration is achieved for the D31 data set: a speed up of 21.51 times better than the ET in CPU.

In this version we worked with synchronous kernel functions since, in the execution of the algorithm implementation in GPU, a kernel function must finish its computation before the other one begins. Consider, for example, the kernel function `EDMGPU` that returns the distance matrix D; such a matrix is an input parameter for the kernel function `KCentresGPU`. Similarly, the computation of the matrix of maxima M, that is returned by the kernel function KCentresGPU, is an input parameter for the kernel function `findminimumofmaximaGPU`. In spite of this synchronism, the implementation of the algorithm in GPU achieves better elapsed times than the implementation in CPU for all the five selected data sets.

Figure 1 shows the elapsed times obtained with the implementations of the algorithm in both GPU and CPU. It is remarkable how the implementation in GPU improves the response times of the algorithm. Clearly, the implementation in GPU exhibits a high speed up for large data sets: those having more than 1000 objects.

[1]These data sets are publicly available from `http://cs.uef.fi/sipu/datasets/`

4 Conclusions

The analysis of the algorithm is fundamental to find the dependencies of the functions on the structures of shared data as well as to be able to take decisions about which portions of the code are likely to be parallelized and which ones are not. Moreover, the design of a parallel algorithm is a challenge because, in order to achieve good accelerations with respect to the sequential algorithm, it must be very clear to the programmer how each architecture internally works such that (s)he is able to take advantage of their individual benefits.

The interaction of the functions over heterogeneous architectures implies, during the programming process, the identification of the portions of the code that can be either synchronously or asynchronously executed. In this way, it is possible to define which functions can be concurrently executed in both CPU and GPU. The programmer is prone to fall into the error of allowing that several processes write on the same position of memory. The question about which data should be written in the shared structure must be carefully solved.

5 Future Work

Although this implementation of the KCentres clustering algorithm presents an acceptable speed up, we consider that it can be significantly improved. Among the possibilities for improvement, it would be convenient to enhance the implementation of the part of the algorithm that searches the minima within the matrix of maxima by using more efficient techniques such as reduction [7], sorting [8] or some other new ones based on the handling of the GPU at the warp level as proposed in [2]. We need to compare the KCentres algorithm performance against other methods like the one proposed in [5] that uses the same datasets.

Acknowledgement

The first author acknowledges funding provided by Universidad Nacional de Colombia through “Convocatoria para la Movilidad Internacional de la Universidad Nacional de Colombia 2017-2018”. Center of Informatics and Systems of the University of Coimbra (CISUC) is also acknowledged.

References

[1] Pavel Paclík Elzbieta Pekalska, Robert P.W. Duin. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2): 189–208, 2006.

[2] Minquan Fang, Jianbin Fang, Weimin Zhang, Haifang Zhou, Jianxing Liao, and Yuangang Wang. Benchmarking the GPU memory at the warp level. *Parallel Computing*, 71:23 – 41, 2018.

[3] Reza Farivar, Daniel Rebolledo, Ellick Chan, and Roy Campbell. A parallel implementation of K-means clustering on GPUs. In *Proceedings of the 2008 International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2008*, pages 340–345, 01 2008.

[4] Alicia Fernández, Álvaro Gómez, Federico Lecumberry, Álvaro Pardo, and Ignacio Ramírez. Pattern recognition in Latin America in the “Big Data” era. *Pattern Recognition*, 48(4):1185 – 1196, 2015.

[5] Marek Gagolewski, Anna Cena, and Maciej Bartoszuk. Hierarchical clustering via penalty-based aggregation and the genie approach. In *MDAI 2016*, pages 191–202, Cham, 2016. Springer International Publishing. ISBN 978-3-319-45656-0.

[6] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010.

[7] Yatong Jiang, Seungmin Rho, Yingping Zhang, Feng Jiang, and Jian Yin. Multi-scale stream reduction for volume rendering on GPUs. *Microprocessors and Microsystems*, 47:133 – 141, 2016.

[8] Erik Sintorn and Ulf Assarsson. Fast parallel GPU-sorting using a hybrid algorithm. *Journal of Parallel and Distributed Computing*, 68 (10):1381 – 1388, 2008.

Paper:#23

Application of active learning metamodels and clustering techniques to emergency medical service policy analysis

By: Antunes, F. Ribeiro, B. Pereira, F.

Application of active learning metamodels and clustering techniques to emergency medical service policy analysis

Francisco Antunes[1]
fnibau@uc.pt
Bernardete Ribeiro[1]
bribeiro@dei.uc.pt
Francisco Pereira[2]
camara@dtu.dk

[1] University of Coimbra
Coimbra, Portugal
[2] Technical University of Denmark
Kongens Lyngby, Denmark

Abstract

Due to the multiplicity of variables and relationships, as well as their stochastic nature, the majority of the real-world urban and transportation systems are not easily modeled by conventional analytic methods. Hence, simulation modeling is a recurrent approach which aims to mimic such environments in order to understand and thus predict their behavior. Nevertheless, in certain extreme scenarios where each simulation run proves to be computationally expensive, detailed exploration of the simulation input space can be compromised or even virtually impossible. Simulation metamodels, along with active learning procedures, have been employed to address such kind of shortcomings.

In this paper, and within the context of active learning, we apply the Gaussian Process (GP) framework as a simulation metamodeling tool to the exploration process of a simulator's output behavior with respect to its inputs variables. The output values predicted by the GP are then clustered so that policy-relevant simulation input regions can be easily identifiable.

We illustrate our methodology using an Emergency Medical Service (EMS) simulator and ultimately analyze the performance of two associated emergency policies. Two output indicators are studied and compared, namely, the average survival rate and the response time. The presented results show that the combined usage of active learning, simulation metamodeling and clustering techniques is able to identify important policy-sensitive regions within the input space of the simulator in study.

1 Approach

The general framework adopted in this work is depicted in Figure 1. It is essentially divided into three parts. The first part encompasses the approximation of the function inherently defined by the simulation model using a GP [4] as metamodel. The fitting procedure is conducted in part two. Both of these parts are iteratively and alternatively employed, eventually defining the procedure we call active learning [5] simulation metamodeling [2]. Here, the use of a GP-based metamodel is meant to avoid exhausting and systematic simulation runs.

The basic idea is to start off with an initial and relatively small training data set, which consists of simulation results, i.e., input-output tuples. A GP is fitted to this data and then predictions are made over a predefined simulation input region in which we aim to focus our simulation analysis. Due to the Bayesian properties that characterize the GP framework, each of its predictions is provided in the form of Gaussian distributions, whose variance encodes the uncertainty associated to such estimates. In each iteration, a new GP model is fitted to the current labeled set via maximum (log) likelihood maximization. Afterwards, the trained GP is used to make predictions over the unlabeled set. Then, the active learning scheme queries the top highest predictive variance data points to be labeled by the simulation model. Here, we assume that these points are potentially the most informative ones and from which the GP can learn in a faster and thus more efficient way. After being labeled, the previously selected points are added to the labeled set, naturally resulting in its expansion. This process is iteratively repeated until a certain variance-based criterion is satisfied. In our case, the active learning metamodeling process stops when the total predictive variance over the unlabeled set is reduced by 90% with respect to the total predictive variance of the initial data set.

Finally, after a fitted GP is obtained, part three is dedicated to the meta-simulation analysis where clustering is applied over the predicted simulation output values in order to find policy-sensitive regions as presented in the following sections.

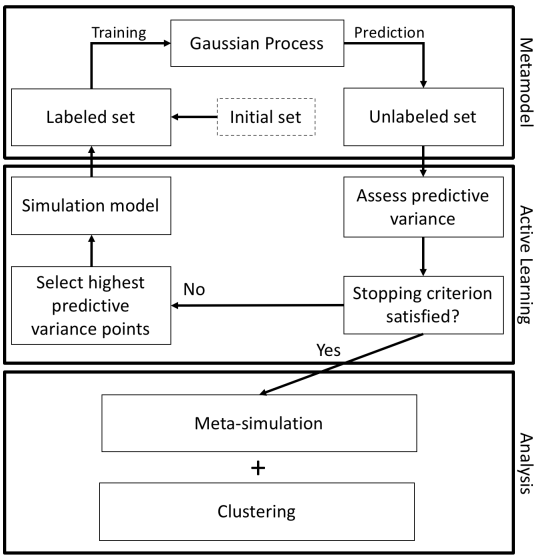


Figure 1: Active learning simulation metamodeling framework with clustering.

2 Experiments

In this work, the Emergency Medical Service (EMS) simulator developed by [1] was studied as a way to illustrate our approach. This simulator is an agent-based model controlled by an emergency medical service agent which allocates and dispatches emergency vehicles (e.g., ambulances). Traffic stochasticity (or error), as well as the EMS vehicles movements, are simulated via a network agent. On the other hand, emergency events are agents that represent the EMS calls and are induced according to a historical database and to a location change probability. If, on the one hand, the traffic stochasticity encodes the difficulty of predicting the network traffic congestion, on the other hand, the probability of location change tries to emulate the inherent dynamics of emergency calls and events. These two important variables should be carefully taken into account by the emergency service operator, so that poor vehicle allocations and dispatching decision are avoided and therefore fatal occurrences (due to delayed assistance) are minimized.

With respect to emergency medical services, the evaluation of policies or operational decisions is virtually impossible in practical terms. Optimization tools, along with simulation models, are usually preferable to emulate the real-world systems as they are able to provide important performance indicators that assess the implemented decisions. In this work, the simulation model numerically computes EMS solutions and reports the corresponding performance metrics in terms of the average vehicle response time and victims' average survival rate. Moreover, this simulator takes into account three types of inputs, namely the emergency (call) location chance probability, traffic error and station locations. In total, the number of inputs sums up to 92. The possible values for the first two inputs lie in $[0,1]$, whereas the location-related inputs assume discrete positive values that represent not only the emergency station location but also the assigned number of vehicles to each of one them.

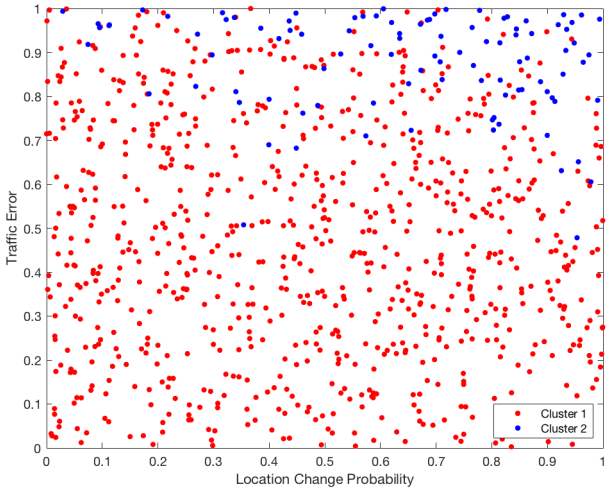
In [3], a study was conducted with a group of level 1 trauma patients in order to assess the 8 minute rule for ambulance response and its impact on the survival rate. It was concluded that there were no statistically significant differences, with respect to the survival rate, between victims that were assisted in and above the mentioned emergency response time

policy. In the same work, it is reported that the survival rate is 0.19 for response times above 480 seconds. Using the earlier mentioned EMS simulator and taking these thresholds into account, we cluster the results provided by the GP-based meta-simulation analysis in order to observe in which combinations of the inputs values such emergency policy requirements are met or not. The K-means was employed with $k = 2$.

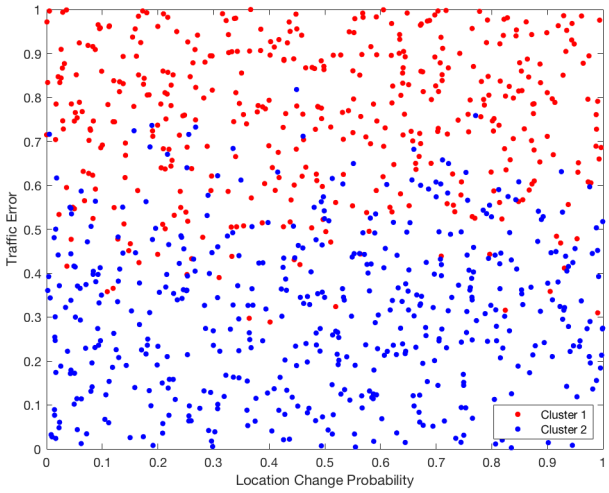
3 Results

The clustering results are depicted in Figure 2. Due to paper space constrains, we only present the results for two simulation input dimensions. In Figure 2(a) the red dots (cluster 1) represents those observations whose average survival rates are greater than 0.19. Contrariwise, the blue dots (cluster 2) correspond to those instances for which the same rate is lesser then or equal to 0.19. Thus, we can see that the most fatal cases are mainly concentrated on the top-right corner, roughly within the rectangle defined by $[0.3, 1] \times [0.6, 1]$. It is clear that as the location change probability increases the number of low survival rates start to increase too. A similar behavior is observable when the traffic error increases. These results confirm our initial guesses with respect to the general behavior of the simulator. As the traffic prediction errors increase, we should expect that more inadequate operation decisions can be taken, eventually leading to a higher number of fatal situations. On the other hand, increasing the location change probability means that the emergency events will present more location variability when compared against the historical data.

In Figure 2(b) presents the clustering results for the average response time. Here, cluster 1 (red dots) represent those situations where the dispatched emergency vehicle time is equal to or greater than 480 seconds, whereas cluster 1 (blue dots), groups the complementary observations. The observations associated to higher response times are essentially concentrated in $[0, 1] \times [0.6, 1]$. Moreover, we can also see that not all of these observations are associated to survival rates below 0.19. This meets the conclusions drawn in [3] that response times below 480 seconds (8 minutes) are not necessarily associated to higher survival rates. Once again, we observe that as the location change probability increases, the response times also increase. This results meets our intuition. The same it true for the traffic error input variable. However, it is interesting to notice that, in this case, the traffic error seems to have a particularly significant role when compared with the location change probability. As the traffic errors increase we can clearly observe that the vehicles delays, caused by unexpected traffic congestion, start to appear. On the other hand, for a given traffic error value, the location change probability does not seem to have a significant impact on the number of delays above 480 seconds.



(a)



(b)

Figure 2: Clustering results for (a) average survival rate and (b) average response time.

4 Conclusions

This paper presented a framework based on active learning metamodeling that allows for an efficient exploration of a simulation behavior. Moreover, a clustering scheme was used over the predicted simulation results in order to provide an easy way to identify regions of the simulation input space that may lead to success, or lack of it, of certain policies and decisions.

In the context of EMS simulation, we illustrate our approach using two important emergency thresholds values, namely, 0.19, for the average survival rate, and 480 seconds, for the average response time. The obtained clustering results showed that, contrary to what we could expect and for the EMS simulator in question, is it not true that higher response times lead to higher mortality rates. The two studied simulation inputs, traffic errors and location change probability seem to have an interesting impact on the output system performances.

This work can be improved in several ways. The presentation of higher-dimensional input regions is an important challenge for the future. During the metamodeling stage, we also did not used any particular experimental design (e.g., Latin hypercube). More complex policies as well as more input dimensions should be considered in the future. Other clustering techniques should also be applied and compared.

Acknowledgments

The portuguese funding agency for science and technology (FCT, I.P.) is gratefully acknowledged for financing the first author with grant No. PD/BD/128047/2016.

References

[1] Marco Amorim, Sara Ferreira, and AntÃşnio Couto. Emergency medical service response: Analyzing vehicle dispatching rules. *Transportation Research Record: Journal of the Transportation Research Board*, page 0361198118781645, 2018. doi: 10.1177/0361198118781645.

[2] Linda Weiser Friedman. *The simulation metamodel*. Springer Science & Business Media, 2012.

[3] Peter T Pons and Vincent J Markovchick. Eight minutes or less: does the ambulance response time guideline impact trauma patient outcome? *The Journal of Emergency Medicine*, 23(1): 43 – 48, 2002. ISSN 0736-4679. doi: https://doi.org/10.1016/S0736-4679(02)00460-2.

[4] C. E. Rasmussen and C. Williams. *Gaussian processes for machine learning (Adaptive computation and machine learning)*. The MIT Press, 2005.

[5] Xizhao Wang and Junhai Zhai. *Learning from Uncertainty*. CRC Press, 2016.

Paper:#24

Characterization of the Human Gait through a Pressure Platform

By: Bastos, M. Coutinho, F. Tonelo, C.

Characterization of the Human Gait through a Pressure Platform

Maria Inês Bastos[1]
a21250160@alunos.isec.pt

Fernanda de Madureira Coutinho[1,2]
fermaco@isec.pt

Cláudia Tonelo[3]
claudiatonelo@sensingfuture.pt

[1] Coimbra Polytechnic - ISEC
Rua Pedro Nunes Quinta da Nora, 3030-199 Coimbra, PT

[2] Institute of Systems and Robotics - University of Coimbra
Rua Sílvio Lima- Polo II, 3030-290 Coimbra, PT

[3] Sensing Future Technologies - Instituto Pedro Nunes
Bloco C, Rua Pedro Nunes, 3030-199 Coimbra, PT

Abstract

Human stride and gait analysis have increasingly become the focus of research studies, giving rise to the development of technological solutions that help measuring its parameters. The PhysioSensing system is one such solution, which is composed of a pressure platform and a computational application that provides visual biofeedback. The purpose of this work is to enhance this system by providing features for aiding the healthcare professional performing dynamic gait analysis. Four prototypes were implemented (in C# and WPF) supporting four dynamic gait analysis profiles: Diabetic Foot Analysis, Footfall Analysis, Asymmetries Analysis, and General Analysis. These prototypes create clinical reports at the end of each session, and were successfully validated in tests conducted in a Clinic of Physical Medicine and Rehabilitation.

1 Introduction

Human gait is characterized by a complex set of movements and the fact that these vary from individual to individual makes its analysis difficult and challenging.

Traditionally, the approach taken by health professionals for gait analysis was based primarily on visual observation. However, given the limitations of this approach, there has been a growing effort and interest of the scientific and business communities to build prototypes and present equipment that will assist the health professional in this task, namely in the recognition of parameters and gait patterns of their patients. These often use a platform of pressure sensors to perform static plantar and postural analysis. However, static analysis is of limited usefulness in tasks such as footfall analysis or plantar ulcer risk prediction, which require consideration of *dynamic* gait parameters.

The address this, the main objective of this work was enhancing a commercial equipment - PhysioSensing - marketed by the Sensing Future Technologies company - with the capabilities to perform the dynamic gait analysis.

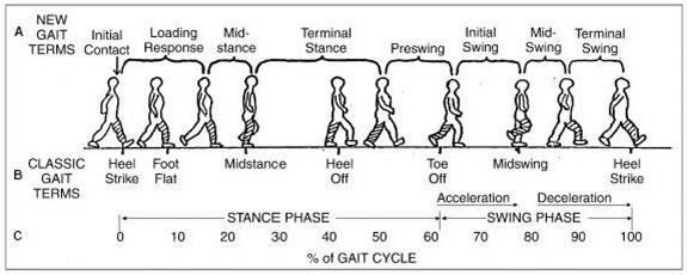


Figure 1: Walking cycle (Source: [1]).

1.1 Gait Cycle

A gait cycle starts with the touch of one of the heels on the ground and ends when that same heel touches the ground again. The gait cycle is divided into two main phases: stance phase and swing phase, which represent respectively 60% and 40% of the cycle [1].

Figure 1 shows the various phases of the gait cycle according to the two most commonly adopted terminologies. The classical terminology of gait recognizes six phases: heel strike, foot flat, midstance, heel off, toe off and midswing. An alternative and more recent terminology recognizes eight phases: initial contact, loading response, midstance, terminal stance, preswing, initial swing, midswing and terminal swing [1].

1.2 Plantar pressure study parameters

The parameters of plantar pressure analysis published in the literature and used in this work were the following:

i. **Pressure map** - is constructed based on the values of maximum pressure acquired in each sensor, presenting them in the form of a map with a color scale [2] (Figure 2).

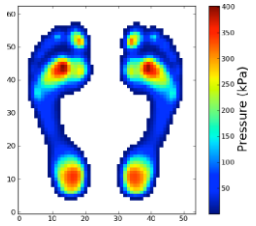


Figure 2: Pressure map (Source: [3]).

ii. **Mean pressure curve and peak pressure curve** - result from linear interpolation of successive mean and maximum plantar pressure values, respectively, during the entire support phase [4].

iii. **Line of center of pressure (CoP) progression**[1] - results from the linear interpolation of a set of pressure center coordinates acquired during the last stage of support [5].

iv. **Speed curve as a function of CoP** - linear interpolation of successive CoP velocity values; The velocity is obtained based on the mediolateral and antero-posterior displacement of CoP divided by the sampling rate of the sensors [6].

v. **Center of pressure excursion index (CPEI)** - defined in percentage as the ratio between the excursion value of the pressure center (distance from the gait line to the construction line, delineated between the first and last pressure center value recorded in measurement) and foot width [7].

2 Setup

The equipment PhysioSensing, used in this work, has a pressure platform consisting of 1600 pressure sensors, with 1 cm^2 of area each [8]. The sensor data were acquired with a sampling rate of 28.57 Hz. Software development was done using WPF and C # language.

3 Analysis Profiles

By analyzing the typical use of the metrics described in Section 1.2, four different metric profiles, suited to specific dynamic gait analysis tasks, were identified. Prototype support for these profiles was implemented. These are:

- **General Analysis** - gathers the most relevant metrics generically useful for the study of gait, these being the pressure map, containing the gait line, and some functions derived from it, namely the replay function (repetition of the plantar maps acquired during the past) and rollover (acquired plantar maps arranged frame by frame), as well as 3D pressure map visualization. In addition to these functionalities, the average pressure curve, the peak pressure curve and the velocity curve as a function of the CoP are also collected, all of them with an indication of the minimum, average and maximum values obtained.
- **Diabetic foot analysis** - focuses on the pressure map and the peak pressure curve, in order to verify if the peak pressure values in the ante-foot and back foot exceed the established limits (based on [4]), identifying thus possible zones of risk of developing plantar ulcers in diabetic patients.
- **Footfall analysis** - includes the pressure map, containing the gait line, and information regarding the length and width of each foot as well as the corresponding CPEI value. Through the values obtained for the CPEI it is possible to identify the type of footfall performed, which can be pronated, normal or supinated, based on values pointed out by [9].
- **Asymmetries analysis** - includes the most relevant metrics for the detection of asymmetries in the distribution of loads between the

[1] Also known as gait line.

two feet, namely the pressure map and the graph of the mean pressure curve, aiding the precaution of plantar lesions and the correction of pathological walking pattern.

The software was complemented by a connection to the database of the current system and the implementation of a functionality that allows clinical reports to be generated at the end of each session.

4 Experimental Results and Discussion

Preliminary usability tests for each of the four prototypes were performed at a Clinic of Physical Medicine and Rehabilitation, based on a sample of ten users (20% female and 80% male; average age 54.1 with square deviation (SD) of 16.9; average weight of 79.7 with SD of 13.8), with several types of pathology, mainly associated to the lower limbs, namely sprain with fracture, knee prosthesis, fracture of the two feet, bi-lateral gonarthrosis, among others. Dynamic gait parameters are collected as the user steps into the sensing platform.

The results, for each one of the four prototype support systems (Section 3), are briefly presented:

- **General analysis**

It has been found that the values obtained for the mean pressure and the peak pressure are, respectively, between 0.06 [N/cm^2] and 76.27 [N/cm^2], and 0.07 [N/cm^2] and 93.96 [N/cm^2]. These values are relatively low when compared with those published in the literature [4], for the same measurements. In relation to the values for the CoP velocity, they were found to be between 0 [m/s] and 5.97 [m/s], with their mean values being in the expected range [6].

- **Diabetic foot analysis**

It was observed that none of the users exceeded the values set for the peak pressure in the right and left foot. However, there are two cases in which the value stipulated for the peak of pressure in the right forefoot is exceeded, as well as for the left foot, in which there were also two cases in which the values obtained are higher.

- **CPEI analysis**

Once the CPEI prototype was analyzed, it was verified that for the right foot the predominant type of footprint is the pronated (result of 6 users), followed by the normal footprint (3 users) and the supinated one (1 user). For the left foot the results undergo some changes, with the normal footprint being the predominant one (result of 7 users), followed by the pronated footfall (2 users) and the supinated one (1 user).

Figure 3 shows the example of 3 users with each one of the three scenarios.

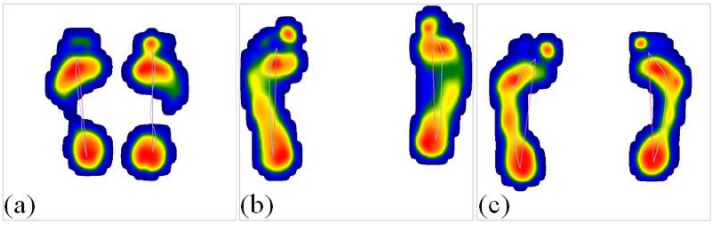


Figure 3: Pressure maps of 3 users for the CPEI prototype. (a) Pronated footprint; (b) Normal footprint; (c) Supine footprint.

- **Asymmetries analysis**

Figure 4(a) shows the pressure map and Figure 4(b) the mean pressure curve for a patient of 63 years old, 98 kg, 1.72 m and with a prosthesis in the left knee. The wearer exerts more pressure on the forefoot area (about 26 [N/cm^2]) when compared to the other areas (below 2 [N/cm^2]). Figure 4(c) shows the pressure map and Figure 4(d) the mean pressure curve for a 26-year-old male, 62 kg, 1.83 m, who fractured both feet about a year and a half ago. The patient presents similar pressure values for both feet, presenting no symptoms of asymmetries.

5 Conclusions

The challenge behind this work was to develop a software package capable of performing the dynamic gait analysis and integrating this functionality into an existing commercial system that is currently only capable of performing the static plantar and postural analysis. The validation of the developed work was carried out with a sample of 10 patients who suffered from several pathologies.

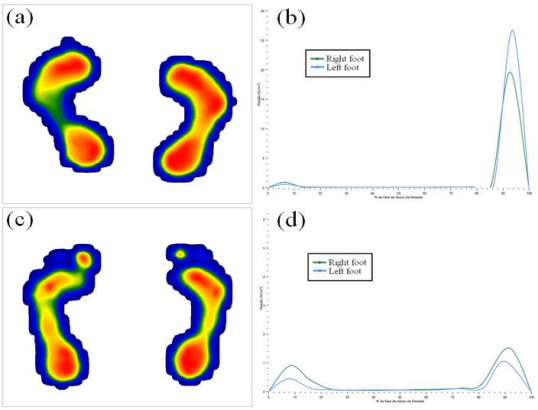


Figure 4: Pressure maps and average pressure curves of 2 patients for the asymmetries prototype. (a) Pressure map and (b) pressure curve for patient 1; (c) Pressure map and (d) pressure curve for patient 2.

The values obtained in the tests are mostly in line with expectations and data found in published literature, suggesting that the system is able to provide good support for dynamic gait analysis. Extended validation efforts with a larger and more diversified sample, and increased participation of healthcare professionals should follow.

Analysis of the results also suggests the consideration of promising alternative metrics, such as using the area value between the two lines instead of the distance value in the CPEI calculation.

Adjustments to hardware parameters such as frequency of acquisition of sensor data, and the area and type of sensors on the platform will also be considered.

It was concluded that, using a platform of pressure sensors, it was possible to obtain satisfactory results regarding plantar pressure parameters and that will certainly help the health professional to analyze the gait dynamics.

References

[1] Physiopedia, "Gait". https://www.physio-pedia.com/Gait.

[2] S. Karki, J. Lekkala, T. Kaistila, H.J. Laine, H. Maenpaa, and H. Kuokkanen, "Plantar pressure distribution measurements: an approach to different methods to compute a pressure map", *Fundamental and Applied Metrology*, vol. 23, pp. 1770-1774, 2009.

[3] S. Karia, S. Parasuraman, M. Khan, I. Elamvazuthi, N. Debnath and S. Ali, "Plantar pressure distribution and gait stability: Normal vs high heel", in *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, 2016.

[4] C. Giacomozzi and F. Martelli, "Peak pressure curve: An effective parameter for early detection of foot functional impairments in diabetic patients", *Gait&Posture*, vol. 23, no.4, pp. 464-470, 2006.

[5] M.C. Chiu, H.C. Wu, L.Y. Chang and M.H. Wu, "Center of pressure progression characteristics under the plantar region for elderly adults", *Gait&Posture*, vol. 37, no.3, pp. 408-412, 2013.

[6] M.W. Cornwall and T. McPoil, "Velocity of the center of pressure during walking", *Journal of the American Podiatric Medical Association*, vol. 90, no. 7, pp. 334-338, 2000.

[7] M.A. Diaz, M.W. Gibbons, J. Song, H.J. Hillstrom and K.H. Choe, "Concurrent validity of an automated algorithm for computing the center of pressure excursion index (CPEI)", *Gait&Posture*, vol. 59, pp. 7-10, 2018.

[8] Sensing Future Technologies. http://www.physiosensing.net.

[9] H.B. Menz, A.B. Dufour, J.L. Riskowski, H.J. Hillstrom and M.T. Hannan, "Planus foot posture and pronated foot function are associated with foot pain: The Framingham foot study", Arthritis Care Res (Hoboken), vol. 65, no. 12, pp. 1991-1999, 2013.

Paper:#27

Automatic evaluation of ERD in e-learning environments

By: Lino, A. Rocha, A. Macedo, L.

A virtual learning environment to evaluate entity relationship diagram automatically

Adriano Lino
adrianolino@dei.uc.pt
Álvaro Rocha
amrocha@dei.uc.pt
Luis Macedo
lui@dei.uc.pt

CISUC
Department of Informatic Engineering
University of Coimbra
Coimbra, Portugal

Abstract

Currently, computer-aided assessments are widely used for multiple-choice questions, but they do not have the ability to assess student knowledge more comprehensively, going beyond right or wrong, which is necessary in teaching about constructing entity relationship diagrams.

This article presents a novel approach for automatic evaluation of entity relationship diagrams that returns in addition to a response in the correct result, a grade that most closely approximates the optimal solution. The method, based on machine learning techniques, uses structured query language metrics extracted from the entity relationship diagram and expert grade to create the prediction model. The preliminary results show our approach as an alternative to automatically evaluate entity relationship diagrams.

1 Introduction

In education, diagrams are explored as tools that enhance learning [1]. Especially when dealing with entity relationship diagrams (ERD), the ERD evaluation offers challenges that attract the interest of the research in the automatic evaluation of diagrams. Those challenges are: a) Type of question: Because of the intrinsic ambiguity of the question, a wide range of questions can be used and this provides a comprehensive scope of ERD assessment; b) ERD notations: The basic notation of ERD is extensible to include concepts of different notations to design more complex structures and it can be used in several contexts in database teaching (DB); and c) Popularity: The entity relationship model (ERM) and its extensions are widely used in industry and academia.

This article presents the first experimental results of application of an approach for automatic evaluation of ERD through SQL metrics and supervised machine learning, which assigns a grade in the diagram presented by the student, which is the student's distance to the optimal solution initially registered by the professor. When the student makes a superior solution, this diagram automatically becomes the ideal solution. It also presents the preliminary results obtained from the first interaction of the design science research (DSR) method and the knowledge built up to date.

This first experiment attempts to prove the hypothesis that the evaluation of a professor's ERD can be predicted by a supervised Machine Learning (ML) algorithm. Preliminary results indicate that artificial neural networks (ANNs) [3] and genetic symbolic expression (GEP) [2] algorithms present good accuracy and are an alternative for the automatic evaluation of ERD. Still the results include the limitations of this approach and suggest future work.

Beyond this introduction, the paper is organized in the following sections: 2) Features, 3) Dataset, 4) Preliminary Results, 5) Conclusions and the future work.

2 Features

The experiment based on machine learning technique uses structured query language features extracted from entity relationship diagram and expert grade to create the prediction model.

The features were defined based on a literature search that aimed to identify the ways to represent ERD concepts in metrics. The metrics are used to extract the information from the diagrams, both in the student's response and in the model response. The answer model is the answer extracted from the book that has the maximum grade. The other responses are random variations of the model that have been evaluated by experts.

The premise for choosing features is to represent the main concepts used in the teaching and learning process of ERD, that is, the features were defined from the direct relationship between the concepts addressed in the teaching of ERD and its representation through metrics [4].

Among these concepts, we highlight the primary keys, the foreign key, and the types of relationships, as the main attributes to be extracted from the ERD. From this, 13 features were selected, 11 of which were inspired by the work of Piattini [4], which defined a set of metrics to measure the quality of an ERD, and two related to the ERD grade (see Table 1). The features used were selected from a review on measurement of ERD.

Table 1. List of entity relationship diagram features.

Metric	Description
NT	Number of entities or tables
NC	Number of attributes or columns includes entity and relationship attributes
NR	Number of relationships
RU	Number of unary relationships (degree)
RB	Number of binary relationships count (degree)
RT	Number of ternary relationships (degree)
RMN	Number of relationships M:N (cardinality)
R1N	Number of relationship 1:N (cardinality)
RNAry	Number of N-Ary relationship (cardinality)
RR	Number of reflexive relationships, (recursive)
RISA	Number of ISA relationship
CX	Grade from experts
RX	Predicted grade

3 Dataset

The project is strongly associated with the academic universe, and therefore the data to be used should represent this universe, rather than industry or business. Therefore, all data were collected from exercises and examples from 18 books.

These books were selected from within a total of 49 most cited books in DB courses in google scholar as well as the books adopted by higher education institutions and also indicated by the ACM and IEEE in their curriculum guides of the courses in computer science and technology. A total of 49 books from the DB and ERD area were found, ranging from 1985 classics to the most recent book publications of the year 2016 (see Figure 1). However, many of these books were discarded for not addressing the topic of ERD in whole or in part. The exclusion criteria of the books were the lack of diagrams illustrating ER concepts, the use of a single study for all ER concepts, the exclusive use of the SQL language to exemplify ER concepts, and no list of exercises with answers or teacher support. At the end, about 18 books from the DB area were selected to create the dataset with ERD.

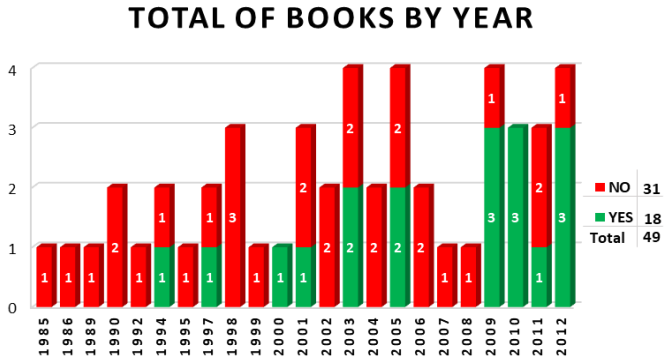


Figure 1. Number of books found about ERD and DB subject.

For this experiment, the machine learning features are metrics extracted from the ERD, and the labels are the professor's grade.

The professor usually uses a scale defined by the university to assign the grade of the student's assessment, for example, in Portugal is used the numerical scale of 0 and 20 values (considering that the student who has obtained a minimum of 10 values has been approved), in Brazil some universities use the numerical scale of 0 and 10 (considering that the student has obtained a minimum of 7 values), in India, some universities use the scale in percentage of 0 and 100% (considering approved the student has obtained a minimum of 50%), in Sweden it is used a classification system A-B-C-D-E-F (A-B-C-D-E stands for approval and F stands for reprobation). However, these scales could be converted to each other, following the system criteria for approval or failed student.

We adopted the grade as a continuous variable, on the scale between 1 and 5 (1 - fail, 2 - bad, 3 - satisfactory, 4 - good, 5 - excellent). This scale was used because it has the benefits, a lower number of values to be predicted by the machine learning if compared with the scales from 0 to 10, or 0 to 20 or percentage scales. It can be more easily adapted to categorical classifications and is still universally known.

The dataset has a total of 90 ERDs, which were divided into 40% for the training phase, 30% for the test phase and 30% for the validation phase with two specific supervised machine learning algorithms which we have been using for automatic evaluations SQL programs [3], [5].

4 Preliminary Results

The results of artificial network training can be observed in Table 2. The columns in the table describe the algorithms used, the metrics used to evaluate the model performance, the training, validation and test phases in the case of 20 numbers of hidden layers.

The results obtained by the network training of the algorithms, in the mapping of the mean square error (MSE) and R^2 values, allow to infer that: a) The graded conjugate gradient algorithm obtained the worst results when compared to Levenverg, having as main high characteristic MSE values; b) The Levenberg algorithm has the second best overall performance, since its R values are higher and consequently have greater predictability and lower MSE values, with lower error; c) Both algorithms have R much closer to one, however, only the Levenberg algorithm obtained MSE values that are much smaller, which makes the model much more predictive.

Table 2. Anns models with 20 hidden networks and through the scaled conjugate, levenberg-marquardt algorithms.

Algorithms	Metric	Train 20	Valid 20	Test 20
Scaled Conjugate Gradient [6]	MSE	0.387576	0.727473	0.615979
	R^2	0.876754	0.999546	0.822608
Levenberg-Marquardt [7]	MSE	0.423967	0.295213	0.301883
	R^2	0.859286	0.999408	0.908086

The results of the GEP training can be observed in Table 3. The columns in the table describe the algorithm used, the two metrics used to evaluate the performance of the model, and the largest error detected.

The results obtained by the GEP training, in the mapping of the MSE and R^2 values, allow to infer that: a) The GEP algorithm obtained the worst results when compared to the Levenberg and Scaled Conjugate Gradient algorithms, having as main characteristic the MSE high values and R^2 is lower than the ANNs algorithms; b) The greatest error detected in the model is 17.86%.

Table 3. Symbolic model statistics.

Algorithms	Metric	Results
Symbolic Regression	**Mean square error (MSE)**	**0.618513522**
	R-Square (R^2)	**0.745036649373373**
	Greatest error:	**17.86%**

5 Conclusion and Future Work

This activity of defining a database of questions from books made it possible to develop a conceptual proof about our approach, which reaffirmed our assumptions described in two of our already published approaches, one with the use of symbolic regression [5] and another with the use of ANN [3]. Our set of features was based on the research on the use of metrics to evaluate ERD [4].

The preliminary results present good accuracy for a proof of concept because the predict evaluation model provides close results to human evaluator. These results also presented the same tendency of ANN to have a better generalization capacity of the model about the symbolic regression.

The importance of the features in the generation of the model can be seen in Figure 2, the sensitivity of them presents the limitation of our approach, highlighting the variables with low value as the metric NR. Sensitivity implies negatively the performance of models designed by machine learning algorithms and can be improved. On the other hand, this limitation should be understood as a new research opportunity, in order to find new features and possibly eliminate others that were already indirectly represented from other metrics. Such as the metric number of relationships, which is also replicated in another relationship metrics.

In this sense, future work includes studies to find out which of these metrics can be excluded without negatively impacting the automatic evaluation model to be proposed.

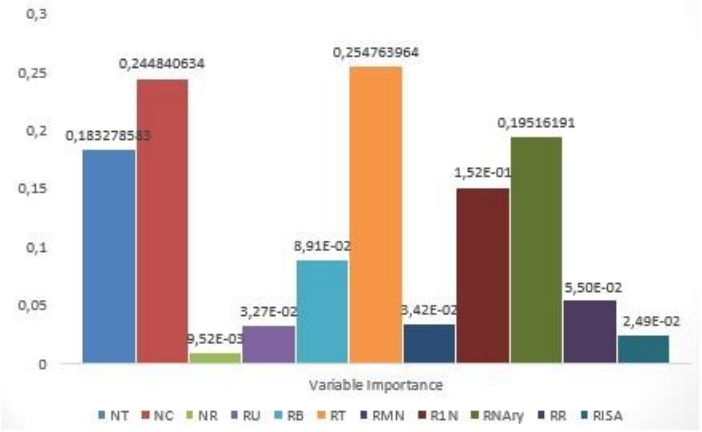


Figure 2. Sensitivity of metrics on the model.

References

[1] P. C.-H. Cheng, R. K. Lowe, and M. Scaife, "Cognitive Science Approaches To Understanding Diagrammatic Representations," in *Thinking with Diagrams*, Dordrecht: Springer Netherlands, 2001, pp. 79–94.

[2] A. Lino, Á. Rocha, and A. Sizo, "A Proposal for Automatic Evaluation by Symbolic Regression in Virtual Learning Environments," in *New Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing*, 444th ed., Springer International Publishing, 2016, pp. 855–865.

[3] A. Lino, Á. Rocha, and A. Sizo, "Virtual teaching and learning environments: automatic evaluation with artificial neural networks," *Cluster Comput.*, pp. 1–11, Sep. 2017.

[4] M. Genero, G. Poels, and M. Piattini, "Defining and validating metrics for assessing the understandability of entity–relationship diagrams," *Data Knowl. Eng.*, vol. 64, no. 3, pp. 534–557, 2008.

[5] A. Lino, Á. Rocha, and A. Sizo, "Virtual teaching and learning environments: Automatic evaluation with symbolic regression," *J. Intell. Fuzzy Syst.*, vol. 31, no. 4, pp. 2061–2072, Sep. 2016.

[6] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.

[7] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

Paper:#28

Evaluation of ensemble methods for predicting defects in sheet metal forming

By: Oliveira, N. Prates, P. Ribeiro, B.

Evaluation of ensemble methods for predicting defects in sheet metal forming.

Nelson Oliveira[1]
njoukov@student.dei.uc.pt
Pedro Prates[2]
pedro.prates@dem.uc.pt
Bernardete Ribeiro[1]
bribeiro@dei.uc.pt

[1] CISUC- Department of Informatics Engineering
University of Coimbra
Coimbra, Portugal

[2] CEMMPRE -Department of Mechanical Engineering
University of Coimbra
Coimbra, Portugal

Abstract

Predicting defects is a challenge in many processing steps during sheet metal forming because a great number of variables is involved in the process. In this paper, an empirical study is presented with the objective to choose the best configuration of an ensemble learning algorithm that will be able to predict sheet metal forming defects. For this purpose, three distinct datasets were generated from numerical simulation results of a sheet metal forming process. Three types of sheet materials were used, one for each dataset: Mild Steel, HSLA340 and DP600. In this work two types of defects, springback and maximum thinning, were considered. The experiments were performed different ensemble learning models using two methods: Stacking and Majority Voting. In this process several Principal Component Analysis (PCA) values for the preprocessing stage and distinct base learners were used. Results show that one ensemble method performs better than the other, one depending on the type of material.

1 Introduction

Predicting defects is commonly applied in the machine learning (ML) field. However, there are seldom applications taking into account sheet forming in the manufacturing process of automative parts. This is a challenge problem because there are many types of conditions that affect the final outcome of the product, leading either to a good final product or a defectuous one, which is considered as a waste to the industry. The application of ML algorithms in this type of problems is useful in order to recognize what are the conditions that affect the final quality of the product. In this line, they can be related to the chemical and mechanical properties of the metal sheet or even to the manufacturing process itself. The fact that the ML technology allows us to predict the final outcome of a metal sheet can help the industry to reduce their waste in terms of time, money and materials. By using multiple mathematical operations that take into account many characteristics of the material and the process, which is impossible to do by a human, ML is no doubt a technique of choice. This is aligned with the idea that if it is possible to previously know (with a good model) if a certain metal sheet within certain conditions of the manufacturing process will have defects, the companies certainly will not buy such metal sheets with the given characteristics.

The focus of this paper is on evaluating how different ensemble methods with different number of base learn classifiers affect the prediction of the model using different PCA values.

2 Sheet Metal Forming Process

Sheet metal forming is a manufacturing process that is widely used in the production of metal components for the most various types of industries. During the process the metal sheets are plastically deformed in a corresponding shape. The forming tools that are usually used are a punch, a die and a blank holder. Generally, sheet metal forming processes allow obtaining high quality components with high cadence and low cost, but however due to the variability inherent to mechanical properties, tool geometry and process parameters makes the formed components often prone to defects such as wrinkling, tearing, excessive thinning and springback. These defects can appear during any phase of this process.

Due to the increasing demand of quality in the the products and the competitiveness among different industries, the Finite Element Method (FEM) is a well-established computational tool that plays a key role in predicting defect-prone regions in components. Recently, some authors integrated statistical descriptions of variability within FEM, for assessing the sensitivity of defect predictions to scatter [1], [2].

3 Proposed Approach with Meta-Learners

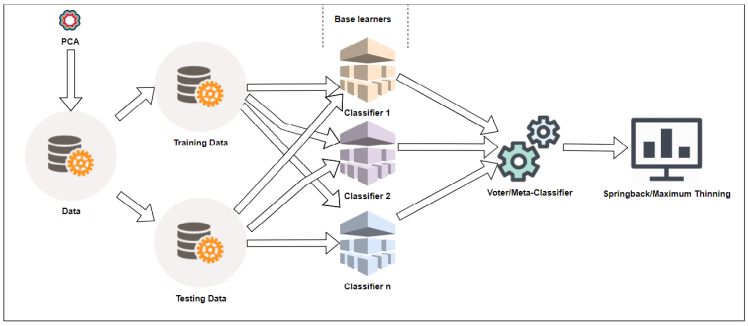


Figure 1: Ensemble model

In the Figure 1 we present the architecture of our model. In a first phase we use PCA to reduce the variance of our data. The next phase consists into selecting n classifiers to be trained with the training data which corresponds to the 70% of the data. In the learning phase, in the case where we use the majority voting method we apply a voting system to the outputs of the previous classifiers to identify if a piece has that defect or not. In the case where we use the stacking method, after the learning phase of the n base learners. It is important to mention that we train the same ensemble separately for each defect, since we assume the Independence between them, and in the end we join the results and convert them in a 4 class evaluation that we explain in the section 4. The above ensemble methods we will tackle in this problem can be summarized as follows:

Majority Voting In majority voting in the first phase each classifier is trained and makes the output prediction. The final prediction is the class that has more half of the votes

Stacking Stacking is an ensemble learning technique that consists in combining many algorithms (base level models) and their outputs are used as features to train another model (meta-classifier or meta-regressor).

4 Experimental Setup

PCA Preprocessing Stage PCA is used to reduce the variance of the data. In our model we've used the following values, 95%, 96%, 97%, 98%, 99% and 100%, the latter meaning that the data stayed as it was originally. Here we aim to investigate if all of the characteristics of the materials and process are relevant to obtain the best possible performance.

Datasets and Features In Table 1 we present the different number of examples for the 3 datasets (materials) of across the 4 classes (defects).

Table 1: Number of pieces per class

Defect \ Material	Mild Steel	HSLA340	DP600
None	27	19	15
Springback	70	68	88
Maximum Thinning	68	73	59
Both	9	10	8

Class **None** corresponds to the pieces that do not have neither a springback defect neither a maximum thinning defect, class **Springback** corresponds

Table 2: Best performances for the Majority Voting Ensemble

Material	PCA preprocessing	Number of base learners	Algorithms	F-score	Accuracy
Mild Steel	99%	3	MLP, Gaussian NB, Logistic Regression	64.92%±9.68%	75.48%±5.52%
HSLA340	100%	5	Random Forest, Decision Tree, KNN, MLP, Logistic Regression	69.8%±8.68%	80.88%±4.95%
DP600	98%	5	Random Forest, Decision Tree, MLP, SVM, Logistic Regression	73.51%±10.11%	82.45%±4.12%

Table 3: Best performances for the Stacking Ensemble

Material	PCA preprocessing	Number of base learners	Algorithms	F-score	Accuracy
Mild Steel	99%	3	KNN, MLP, Gaussian NB Meta-classifier:MLP	63.15% ± 8.82%	74.71% ± 4.89%
HSLA340	100%	5	Random Forest, Decision Tree, KNN, MLP, SVM Meta-classifier:KNN	68.06% ± 8.73%	81.42% ± 4.13%
DP600	98%	5	Random Forest, Decision Tree, KNN, MLP, SVM Meta-classifier:KNN	73.94%±10.49%	83.04%±3.94%

to the pieces that have a springback defect but do not have a maximum thinning defect, class **Maximum Thinning** corresponds to the pieces that do not have a springback defect but have a maximum thinning defect. Finally the class **Both** corresponds to the pieces that have both defects. We can see that for all the 3 materials the most examples are contained in the classes **Springback** and **Maximum Thinning**, and the fewer examples are contained in the classes **None** and **Both**.

The dataset has 16 features (mechanical properties and the strength of the blank-holder), that are obtained from numerical simulations of U stamping profile [3], representing the properties of each material.

For each material 15 features were considered, related to the following mechanical properties, Young modulus(E), anisotropy coefficients (r_0, r_{45} and r_{90}), initial tensile stress-strain data generated from parameters parameters of the Swift's hardening law(Y_0, C and n) and the initial thickness of the sheet (t_0). Additionally one feature related with the stamping process, which is the blank-holder force (BHF) was considered.

Base learners and Meta-classifiers To train our model we've used the following algorithms 7,K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Support Vector Machines (SVM), Logistic Regression, Multi Layer Perceptron (MLP) and Gaussian Naive Bayes, these algorithms are used either in the base learners as in the stacking method. For the base learners we've used all the combinations of these algorithms in groups of 3, 4 and 5. In order to verify how the model behaves with these combinations.

5 Results and Discussion

In Tables 2 and 3 we present the performance (F-score and Accuracy) of the model when using the majority voting and stacking ensemble methods respectively. In each one of the tables we present the best results (based on the F-score values) regarding the combination of different numbers of base learners and the PCA values that we have used. Comparing the performances of the 3 materials a across both tables, we can see that for the DP600, the performance is always greater than for the other 2 materials. Another first sight observation that we have made is that the Accuracy values are almost 10% greater than the F-score values for each material and the standard deviation is also nearly an half of the standard deviation of the F-score results. Since in the end we have 4 classes that resulted from the combination of the presence and absence of the springback and maximum thinning defects, and as we saw before in Table 1, most of the examples are in the classes **Springback** and **Maximum Thinning**. Observing that we can say that the poor performance of the F-score is due to the lack of examples in the classes Both and None since in the number of examples is far from being balanced.

When we compare the performances for the model regarding the majority voting ensemble, for the Mild Steel material we observe that the performance is greater when we use less base learners (3). In the case of the HSLA340 and DP600 materials the performance was greater when we used more base learners (5).

When we compare the performances for the model regarding the stacking ensemble, for the Mild Steel material we observe that the performance is greater when we use less base learners (3). In the case of the HSLA340 and DP600 materials the performance was greater when we used more base learners (5).

If we look at the two ensemble methods that we have used and compare the results across them, we observe that only for the DP600 material the performance is greater when we use the stacking ensemble method. On the contrary for the Mild Steel and the HSLA340 materials the performance is greater when we used the majority voting Ensemble.

For the PCA values that we have used we saw that the best performance is not always found when all of the principal components are used, meaning that not always all information is relevant to correctly predict the final outcome.

6 Conclusions and Future Work

In this work we have presented an analysis on the performance of ensemble methods and number of base learners used when predicting sheet metal forming defects, using different PCA values. The results of the two ensemble method show that depending on the material that is used one of the methods is better than the other and the same relationship stands for the number of base learners used in the each method. The difference between the F-score values and the Accuracy are due the lack of examples of in the classes **None** and **Both**. To solve this we can furtherly generate synthetic samples. Another way to improve the results is to give a greater weight to the classes with fewer examples.

7 Acknowledgements

The authors gratefully acknowledge the financial support by the Portuguese National Innovation Agency through project Safeforming (ref. POCI-01-0247-FEDER-017762), and by the Portuguese Foundation for Science and Technology through grant SFRH/BPD/10165/2014.

References

[1] R. Chiba. Reliability analysis of forming limits of anisotropic metal sheets with uncertain material properties. *Computational Materials Science*, 69:113–120.

[2] L. Nilsson D. Aspenberg, R. Larsson. An evaluation of the statistics of steel material model parameters. *Journal of materials processing technology*, pages 1288–1297, 2012. doi: 10.1016/j.jmatprotec.2012.01.016.

[3] P. Prates M. Dib, B. Ribeiro. Model prediction of defects in sheet metal forming processes. *Communications in Computer and Information Science*, 893.

Paper:#30

Deep Learning for Drug Target Interaction Prediction

By: Coelho, G. Arrais, J. Ribeiro, B.

Deep Learning for Drug Target Interaction Prediction
Portuguese Conference on Pattern Recognition

Guilherme Coelho
Bernardete Ribeiro
Joel P. Arrais

DEI | CISUC
University of Coimbra
Coimbra, PT

Abstract

The discovery of antibiotics was quickly followed by the emergence of bacterial antibiotic resistance. This resistance makes the discovery of new drugs an urgent need. Identifying interaction between known drugs and targets is a major challenge on drug discovery.

Traditionally, the performance of drug target interaction prediction models depends heavily on the descriptors used, and there is no widely agreement on which drug and target descriptors have the best predictive power. This makes the use of traditional machine learning algorithms a rudimentary approach. In this work, to accurately predict new drug target interactions, we developed a deep learning based architecture, capable of understanding, during training, the best descriptors for the classification.

The proposed model reaches an accuracy of 0.90, outperforming current state-of-the- art methods based on shallow architectures. The results obtained suggest that the model could be further used to predict the interaction between drugs and target and the be used on the identification of new leads for drug repositioning.

1 Introduction

One of the biggest medical breakthroughs of the twentieth century was the discovery of antibiotics, immediately followed by the emergence of bacterial antibiotic resistance. So today, more than ever, is necessary to develop new antibiotics [1].

Drug discovery process is very complex, time and money consuming. Furthermore, 90% of the candidate drugs that enter clinical trials, fails to gain regulatory approval [3]. This indicates our actual inability to entirely realize the potential liabilities of candidate compounds. Thankfully, the recent experimental efforts allowed the compilation of public databases, enhancing the probability of developing efficient computational methodologies to improve this process.

Drug repositioning, which is the application of available drugs for treating conditions different from the original treatment purposes, has been proposed as the best alternative to overcome these issues [4] and can be achieved through computational methodologies.

Systematically assess drug target interactions (DTIs) is one the areas that have, recently, witnessed a great improvement, due the amount of data available, and is the basis of rational drug design. Here we proposed a computational approach, based on deep learning architectures, capable of predicting DTIs, using data both from the drug and its target.

Nowadays, many *in silico* approaches have been developed to identify new drug target interaction, applying mainly traditional methods, such as random forest [2] and support vector machine. These are typical feature-based methods, although there is not an global understanding of what features have a strong predictive power. Deep learning could help solving this issue and several studies now show that deep learning is an important approach to consider and explore on the filed of drug discovery.

In fact, deep learning techniques applied in the proteomics field are still few and recent. Despite successful when used, the full potential of deep architectures in the pharmacological field is yet to be shown, meaning there is still room for improvement. Deep architectures describe the data more precisely by encapsulating the most relevant higher-level abstractions, and so could improve state-of-the-art method for the DTI prediction challenge.

2 Methods

2.1 Drug target space

The construction of the total data set included gathering both positive and negative labeled data for drug target interaction. Known positive drug target interaction data, was collected from two different sources: (1) DrugBank and (2) from a previous work on the field, a DTI prediction study by Yamanishi *et al.*. On opposite to the positive data set, which was based on known positive interactions, the negative data set was built by considering DTIs with experimental bioactivity values greater than 10 μM, since for this range of values the DTIs are considered to possess weak binding activity. We ended up with a drug target space containing a total of 26,945 entries, described by 755 features. Descriptors for both drugs and targets were gathered using *PyDPI*.

2.2 Data preparation

After removing all duplicate entries, avoiding redundant information through similarity analysis, all the data was scaled and normalized, since heterogeneous data can harm a deep network, by hampering its convergence.

Feature engineering is, for traditional machine learning pipelines, one the most important task. For DTI problem, this task becomes more challenging since there is not an agreement on what features have the most predictive power. Fortunately deep learning removes this need, because neural network can, without human intervention, understand which features have more predictive power. So, no feature engineering was performed on the data set.

The entire drug target space was splitted into train and test data. While constructing the best model we used 10-fold cross validation as a evaluation strategy, and the test data, never shown to the model, was used to assess the final chosen model.

2.3 Model implementation and evaluation

In order to build, train and test a deep learning architecture we decided to use TensorFlow and Keras, both technologies compatible with Python.

Despite the widely variety of deep architecture, the most suitable one for our data is a deep feed-forward network, since it is numerical and there is not a sequential correlation between the drug target pairs. Furthermore, this kind of architecture have proved to perform well on extracting, sequentially, abstract representations of raw data.

A deep architecture have many parameters that are trained, as well as, parameters that are defined for the architecture and kept until the final model finalization. The second, most be optimized for our data and so, we performed hyper-parameter tuning through a grid-search, to find out which parameters fitted the model the best.

In order to avoid overfitting we applied the *dropout* technique to our model, as a way to prevent model units from co-adapting too much. Besides that, since we are facing a binary classification problem, we performed a threshold evaluation so we could improve our model.

2.4 Measurement of prediction quality

To assess the performance we calculated four commonly used evaluation metrics, area under the receiver operator characteristic curve (AUC), accuracy (ACC), true positive rate (TPR), known as sensitivity or recall and true negative rate (TNR), known as specificity. We also include F_1 score so we could consider both precision and recall.

3 Results and Discussion

3.1 Baseline approach

Our first goal was to define a baseline to serve as a reference point for comparing how well our model is performing. Our baseline was a naive and simple approach to DTI prediction problem. Surprisingly our baseline achieved an AUC of 0.86 on the external data, which is close to the results

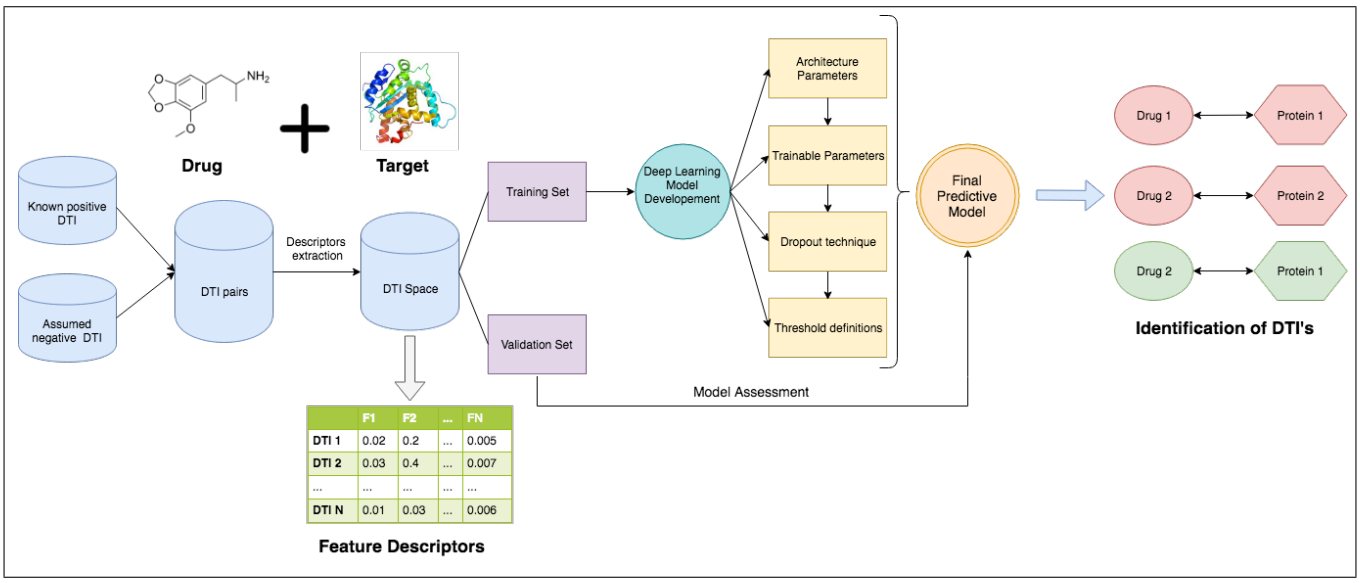


Figure 1: The flowchart of the proposed deep learning pipeline.

Parameter	Value used
Number of Layers	5
Total number of neurons	186
Batch-size	50
Number of epochs	50
Optimizer	adam
Learning Rate	0.001
Momentum	0.4
Weight initialization	'lecun_uniform'
Activation function	ReLU
Loss function	'binary_entropy'

Table 1: Final parameters used on the model constructed.

of other works on the area, aiming for great expectations for the final model.

3.2 Determining deep architecture

The results of an architecture depends heavily on the parameters used. Number of layers and neurons are two of the main parameters to consider. With grid search technique, we manage to find all the best parameters for our data set. On Table 1 we can the group of parameters for which the performance was the best.

Besides all the parameters referred above, which are defined and keep the same throughout the training process, our model is also composed of many other parameters, that are trained on the process. Among all this 26,321 parameters are the weights of each connections between neurons and its biases.

3.3 Network regularization

With many tuning, and adapting all the parameters to the data, the model could be overfitting. As a way to avoid it, we added a dropout layer for the first and second layer of the model, with a rate of 0.2 and 0.1 respectively.

Since we are facing a binary classification problem we decided to inspect the threshold used as a criterion to separate classes. After some manual search, we found out a threshold of 0.8 improved the model. On Figure 1 we can see a structure of the entire pipeline.

3.4 Evaluation metrics

In order to fully assess the performance of the model we calculated mean and standard deviation for evaluation metrics for 100 runs (Table 2).

Achieving 90% of accuracy is a great insight that our model satisfies the goal of correctly predict if a drug and a target will interact. A 99% of specificity proves the model ability to avoid false alarms and to mislead candidates drugs that later will fail, on the drug discovery process.

Evaluation metric	Mean	Standard deviation
Accuracy	0.904	0.003
Sensitivity	0.771	0.008
Specificity	0.990	0.002
Area under the ROC curve	0.880	0.004
F1-score	0.863	0.005

Table 2: Evaluation metrics results for 100 runs.

The overall results obtained suggest that the proposed deep learning architecture could be used in the identification of new leads for drug repositioning, and should be used to improve drug discovery process.

4 Conclusions

Deep learning adoption on biomedicine has been slow and this works intends to contradict this resistance by showing its potential on the field of drug discovery. Furthermore, Big data investments on pharmaceutical industry will reach $4.7 Billion in 2018, which reinforces the need for study and development of novel and better approaches for this field.

Everything indicates that the future of computer-aided drug discovery will be promising. The results obtained here prove that our deep model can be further used to predict DTIs and deep learning will have a major role on this revolution. It is necessary to continue exploring this techniques possibilities and how they could be applied to drug discovery.

Acknowledgments

This research has been funded by the Portuguese Research Agency FCT, through D4 - Deep Drug Discovery and Deployment (CENTRO-01-0145-FEDER-029266)

References

[1] André Santiago José Luis Oliveira António Dourado Joel P. Arrais Edgar D. Coelho, Igor N. Cruz. A Sequence-Based Mesh Classifier for the Prediction of Protein-Protein Interactions. 2017.

[2] José Luís Oliveira Edgar D. Coelho, Joel P. Arrais. Computational Discovery of Putative Leads for Drug Repositioning through Drug-Target Interaction Prediction. 2016.

[3] E. M. Scolnick R. M. Plenge and D. Altshuler. Validating therapeutic targets through human genetics. *Nature Reviews Drug Discovery*, 12 (8):581–594, 2013.

[4] M. D. Krasowski S. Ekins, A. J. Williams and J. S. Freundlich. In silico repositioning of approved drugs for rare and neglected diseases, apr 2011.

Paper:#34

Evolutionary insights from the comparative analysis of hominid genomes

By: Teixeira, A. Pratas, D. Pinho, A. Silva, R.

Evolutionary insights from the comparative analysis of hominid genomes

Ana Teixeira
Department of Medical Sciences, iBiMED

Diogo Pratas
IEETA

Armando J Pinho
IEETA, DETI

Raquel M Silva
Department of Medical Sciences, iBiMED, IEETA

IEETA - Institute of Electronics and Informatics Engineering of Aveiro,
DETI - Department of Electronics, Telecommunications and Informatics,
iBiMED - Institute for Biomedicine,

University of Aveiro

Abstract

Neanderthal groups have been found in Eurasia while Denisovan groups were found only in Denisova Cave (Eurasia) until now. Around 40,000 years ago, these groups inhabited the same places with modern humans. There are many evidences that show genome admixture between these hominid populations. In this study, we use CHESTER to find exact genomic regions (i.e. substitutions are considered absent) present in modern human DNA and not present in ancient genomes. The identification of these regions is important to understand evolutionary traits. We found around 30 coding genes, 6 RNA genes and a few uncharacterized sequences that are present only in modern human DNA. Coding genes are involved in several processes, such as hematopoietic cell differentiation, metabolism and olfactory functions. In the future, the identification and comparison of these sequences is crucial to understand the role of these genes in cell biology, human health and evolutionary history.

1 Introduction

Neanderthals and Denisovans shared a common ancestral population around 381,000 and 473,000 years ago. Neanderthals have been found in Europe and western and central Asia, while Denisovans were found in Denisova Cave (Russia). During the Late Pleistocene period these extinct groups inhabited Eurasia and modern humans spread out of Africa around 350,000 ± 50,000 to 35,000 ± 5,000 years ago (1).

Some studies showed evidences that Neanderthal is the closest hominid group relative to modern humans (2). Traces of their genome were found among all non-African populations, namely, an X-linked haplotype of Neanderthal origin, suggesting that the admixture occurred when modern humans arrived to Eurasia (3) On the other hand, there is scarce information about Denisovans. Previous findings agree that this group contributed to the genomes of present-day Australian Aborigines and American populations. Although small amounts of Denisovan DNA were found, nuclear DNA appears to have higher diversity than Neanderthals, but lower than modern humans (4). Very recently, a bone fragment was found from an individual with a Neanderthal mother and a Denisovan father. The mother was more related with a population found in Croatia than the Neanderthal bones founded in the Denisova Cave, suggesting that the migrations of Neanderthals for all Eurasia occurred before the spread of out-of-Africa by modern humans, as previously suggested (5).

Fossils from ancient hominids are found in relatively close places and several analyses of the relationship between Neanderthals, Denisovans and modern humans showed differences in diversity and population dynamics. Therefore, our objective is to find and identify exact regions in the modern human genome (*Homo sapiens*) that could elucidate past events. Their discovery is important to identify evolutionary traits that could be related to novel functions or related to diseases. For that, we use a probabilistic method (CHESTER) to map and visualize those regions. This is a probabilistic method that is alignment-free with respect to the ancient genome.

2 Method

CHESTER is a method to detect and visualise human specific regions based on the detection of relative absent words (RAWs). RAWs are sub-sequences that do not occur in the reference sequence (given sequence) but do occur in the target (6). In this study we focused on finding RAWs that are present only in modern human and not in Neanderthal and Denisovan genomes (substitutions are considered absent). The implemented fully automatic command line CHESTER (http://pratas.github.io/chester) written in C language, is optimized to work with large ancient genomes. The tool is divided into three programs: CHESTER-map for mapping the regions, CHESTER-filter for filtering and segmenting the regions and CHESTER-visual for visualizing the regions (7, 8).

In this study, we use the high coverage whole Neanderthal genome (raw data), acquired from a Siberian woman toe phalanx bone, found in Denisova Cave (9) and a Denisovan high coverage genome acquired from a phalanx bone and two molars (10) to identify exact regions of the modern human genome. For that, we use the GRCH38p7 reference assembly. The parameters used in this work are described next.

In CHESTER-map, ancient genomes (reference) were used simultaneously against modern human DNA (target), using a Bloom filter with a size of 64 GB (m = 858,993,459,200), a k-mer model with high depth (k = 30) and with the "-*i*" parameter (handle inversions) for the mapping. Reads from reference genomes were used in FASTQ format while reads from target genome were used in FASTA format.

For CHESTER-filter, the obtained coordinates from CHESTER-map were used to filter and segment the exact regions with a window size of 401 and 501bp (base pairs), a threshold value of 0.45 and 0.48 (-*t*) and a subsampling size of 1bp (-*u*). Subsampling refers to the bases that are removed from the output, between each entry.

In CHESTER-visual, the exact regions were visualized with an enlarge (-*e*) of 500,000bp, chromosome by chromosome. The enlarge represents a region that is increased with a certain number of bases for visualization purposes only (7).

After setting the parameters, the tool works with the following algorithm:

- For each ancient DNA sequence:
  - Given m, calculate the number of optimal hash functions (h);
  - Calculate the probability p and the round value of h;
  - Using a window size, the model updates each possible k-mer in the Bloom filter;
  - Freeze the model (stop updating);
- For each human chromosomal sequence, search for each k-mer and for the inverted complemented k-mer, storing the result in a Boolean file;
  - Store the individual results in a global Boolean file;
  - Filter the global file, given a certain window size;
  - Segment the filtered results, according to a threshold t, and store them in a file with their relative genomic coordinates (individual files);
- Finally, read the coordinates of the regions from each file and paint these in an image.

3 Results

In this study, we used Neanderthal and Denisovan genomes as a reference, against the modern human genome (target). We have included the unlocalized ("chr25"), unplaced ("chr26") and mitochondrial ("chr27") sequences, as well as somatic and sexual chromosomes. CHESTER ran with a Bloom filter of 64 GB. The maximum probability of a false positive, p, was 0,000629 and the optimized hash functions was 10. We have split the Neanderthal genome into 5 parts and the Denisovan genome into 6 parts. In our server (Linux with an Intel Xeon CPU E7320 at 2.13 GHz), it took approximately 8 days to run the full experiment (without parallelization).

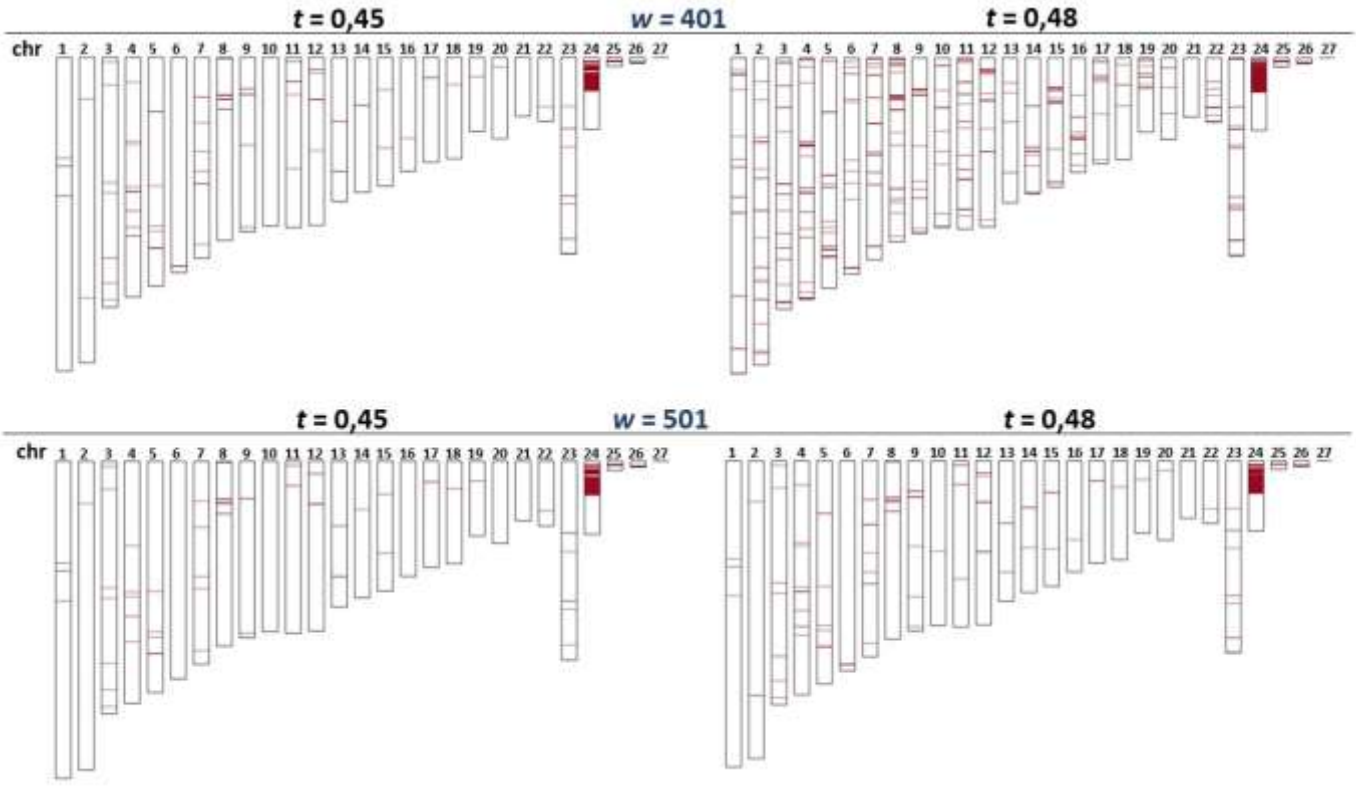


Figure 1: Representation of human specific region maps relative to Neanderthal and Denisovan genomes, by chromosome. Results were obtained using CHESTER with w = 401 and w = 501 with t = 0.45 and 0,48, and k = 30. Red strips represent exact regions.

Results are shown in Figure 1. As ancient genomes were obtained from women, a big number of exact regions in the Y chromosome ("chr24") were found. Therefore, in the total number of regions we did not took these into account. For a window size of 401 and a threshold of 0,45 and 0,48, respectively, we found 150 and 195 exact regions. For the same threshold values but a window size of 501, we found 100 and 172 regions, respectively.

Having the genomic coordinates of the exact regions, we use them to find the corresponding sequence on modern human DNA. For that, we use Ensembl - BioMart (https://www.ensembl.org) and UCSC (https://genome.ucsc.edu) genome browsers. We found around 30 protein-coding genes, around 6 RNA genes, mainly ncRNA (non-coding RNA) and a few uncharacterized sequences. Coordinates with a small and uncharacterized sequences were Blasted to identify the gene (https://www.ncbi.nlm.nih.gov/) (data not shown).

Among protein-coding genes, we have found COL24A1 (collagen type XXIV alpha 1 chain, ENSG00000171502), GHR (growth hormone receptor, ENSG00000112964), CLIC2 (chloride intracellular channel 2, ENSG00000155962) and OR8G5 (olfactory receptor family 8 subfamily G member 5, ENSG00000255298). These genes are involved in several biological processes, such as signal transduction, hematopoietic cell differentiation, hormone response, endocytosis, sense of smell, metabolism and ion transport.

In future work we will look for the genomic alterations between these regions (exact regions vs primitive hominid genomes) to identify its impact in evolution. We will also perform a gene ontology analysis to understand the role of these regions in cell biology and human health.

4 Conclusion

RAWs are small exact regions present in a genomic sequence and their discovery and identification are important to understand evolution. In this study we used CHESTER to find these RAWs in modern human DNA. We found several regions that correspond to around 30 protein-coding genes that are involved in several cell functions. In the future, these regions may be studied to find the exact genomic alterations and its meaning in human evolution.

5 Acknowledgments

This work was partially funded by FEDER (Programa Operacional Factores de CompetitividadeCOMPETE) and by National Funds through the FCT-Foundation for Science and Technology, in the context of the projects UID/CEC/00127/2013, UID/BIM/04501/2013, POCI-01-0145-FEDER-007628, PTCD/EEI-SII/6608/2014, and the grant SFRH/BPD/111148/2015 to RMS.

References

1. Meyer M. Nuclear DNA sequences from the Middle Pleistocene Sima de los Huesos hominins. Nature. 2016;531:504.
2. Pääbo S. Genetic analyses from ancient DNA. Annual Review of Genetics. 2004;38(1):645-79.
3. Yotova V. An X-Linked haplotype of Neanderthal origin is present among all non-African populations. Molecular Biology and Evolution. 2011;28(7):1957-62.
4. Racimo F. Evidence for archaic adaptive introgression in humans. Nature Reviews Genetics. 2015;16:359.
5. Slon V. The genome of the offspring of a Neanderthal mother and a Denisovan father. Nature. 2018;561(7721):113-6.
6. Silva RM. Three minimal sequences found in Ebola virus genomes and absent from human DNA. Bioinformatics. 2015;31(15):2421-5.
7. Pratas D, editor Visualization of distinct DNA regions of the modern human relatively to a Neanderthal genome. Pattern Recognition and Image Analysis. 2017; Cham: Springer International Publishing.
8. Pratas D. Detection and visualisation of regions of human DNA not present in other primates. Proceedings of the 21st Portuguese Conference on Pattern Recognition, RecPad 2015, Faro, Portugal,. October 2015.
9. Prüfer K. The complete genome sequence of a Neanderthal from the Altai Mountains. Nature. 2014;505(7481):43-9.
10. Meyer M. A high coverage genome sequence from an archaic Denisovan individual. Science (New York, NY). 2012;338(6104):222-6.

Paper:#35

Identification of antifungal targets using alignment-free methods

By: Figueiredo, C. Pratas, D. Pinho, A. Silva, R.

Identification of antifungal targets using alignment-free methods

Catarina Figueiredo
IEETA

Diogo Pratas
IEETA

Armando J Pinho
IEETA, DETI

Raquel M Silva
Department of Medical Sciences, iBiMED, IEETA

IEETA - Institute of Electronics and Informatics Engineering of Aveiro,
DETI - Department of Electronics, Telecommunications and Informatics,
iBiMED - Institute for Biomedicine,

University of Aveiro

Abstract

Invasive fungal infections are a growing concern worldwide, especially the emergence of opportunistic pathogen species in nosocomial environments. The increase in *Candida* infections is linked to the ability to colonize implanted medical devices (e.g., central venous catheters) and development of cross-resistance to antifungal compounds that arises mainly due to mutations in the target enzymes or drug efflux-pumps. Here, we have used alignment-free methods with the human genome as reference to screen several yeast genomes and have identified novel targets suitable for the development of antifungal drugs.

1 Introduction

Antimicrobial drug resistance is a global but yet underestimated threat to public health worldwide, compromising the effective prevention and treatment of infectious diseases. Major group risks include immunocompromised patients, namely HIV, cancer, or transplant patients, and even non-immunocompromised hosts such as major surgery or intensive health care unit patients, patients with chronic disease or other comorbidities, newborns and the elderly population.

The high morbidity and mortality rates of fungal infections are due to difficulties in diagnosing and assessing the susceptibility to antifungal agents, as well as easy dissemination of fungal pathogens and rapid development of antifungal resistance [1-6]. In the genus *Candida*, we have previously shown that resistance to different antifungal drugs arises due to mutations in the target enzymes [1-3], drug efflux-pumps [4,5] or the transcriptional factors that regulate their expression [3,6]. The use of drugs with broad antifungal activity for both treatment of patients or as a prophylactic agent, also contributes to the emergence of resistant strains.

Computational tools and methods are useful for the identification of novel antifungal targets, to assist rational drug design and improve current drugs, resulting in better outcomes in fungal infections. To identify potential target regions for novel antifungal drugs, we used alignment-free methods [7,8] to compare the human genome and several *Candida* and other yeast genomes. The results are presented and discussed below.

2 Methods

We have previously developed the tool EAGLE [7] to identify RAWs (relative absent words) in pathogen genomes relative to a host genome. Here, we have applied this strategy to uncover specific regions in fungal chromosomes that are not present in the human genome. These are potential target regions for the development of novel antifungals.

The EAGLE method detects the presence of words with a certain size that are not present in a reference, including the inverted complement of each word. The method uses the notion of absent words [9,10] with particularity for the relative case [7]. The method works as follows. Consider a reference sequence, χ , and η target sequences Y_1, Y_2, ... , $Y\eta$. All sequences are from a finite alphabet, $\sigma = \{A,C,G,T\}$, and $|\chi|$ denotes the size of sequence χ .

We compute the k-mers of χ and the Y_i sequences using a sliding window of size k. Each k-mer is converted into a numeric index, i, and stored into a binary array if k <17, otherwise in a hash table. Parallel, we perform the following mapping: *A*-*T*, *T*-*A*, *C*-*G* and *G*-*C*. The mapping is applied for each reversed k-mer, converted into an index, i, and stored as described above. Each k-mer from χ is loaded, including those from the reverse mapping, and stored in memory. We call this the training phase.

Then we start the matching phase. The intention is to find exact k-mers on each Y_i. Therefore, for each Y_i, a boolean array is created, B_i, with $|B_i|=|Y_i|-k$, containing a true value when a k-mer exists in the memory.

The objective is to detect RAWs. Therefore, the interest is on the false elements from B_i. Since the process of matching is sequential, each position of a false element in B_i reports the exact position in the target sequence, Y_i.

Finally, the results are presented along with each Y_i that depicts the coordinates of the regions or points where the k-mer (absent in χ) occurs.

Accordingly, we use as a training sequence (χ) the human reference genome and each fungi genome sequence (Y_i) as a target. Yeast genome sequences containing all chromosomes for each species were downloaded in FASTA format from the Candida Genome Database (CGD, www.candidagenome.org). The following ten species were considered: *C. albicans* SC5314 (15/04/14), *C. albicans* WO-1 (06/12/13), *C. dubliniensis* CD36 (26/04/13), *C. glabrata* CBS138 (12/07/12), *C. guilliermondii* ATCC-6260 (06/12/13), *C. lusitaniae* ATCC-42720 (06/12/13), *C. orthopsilosis* Co90-125 (06/12/13), *C. parapsilosis* CDC317 (12/03/12), *C. tropicalis* MYA-3404 (06/12/13), *D. hansenii* CBS767 (06/12/13), and *L. elongisporus* NRLL-YB-4239 (06/12/13). EAGLE ran in a LINUX environment with k-mer size set from 10 to 13.

3 Results and Discussion

The results for k=10, k=11 and k=12 are shown in Figure 1. The number of RAWs obtained for k=10 was zero in all genomes, whereas for k=12 and k=13 the number of RAWs identified was too high for further analysis (over 2.000 and 50.000 regions, respectively). Considering k=11, RAWs ranged from 0.0001% and 0.0008% of the genome, which corresponds to 20-100 RAWs in the different species considered (Figure 1).

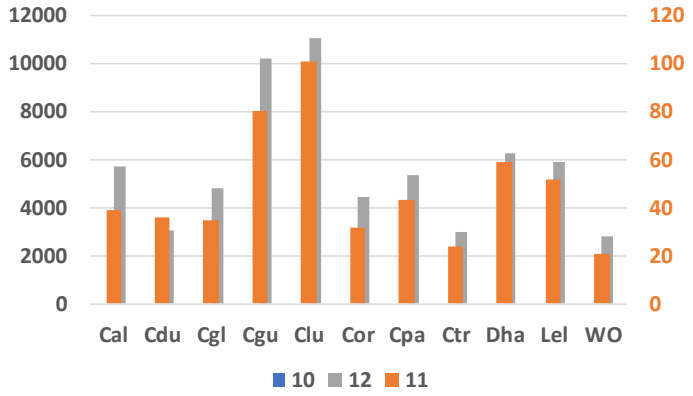


Figure 1: Number of RAWs identified in each genome. For k=10 (blue), no RAWs were identified and for k=12 (grey), the number of RAWs was very high (over 2000 *per* species). Further analysis was performed for k=11 (orange), as described. Cal, *Candida albicans* (39 RAWs); Cdu, *Candida dubliniensis* (36 RAWs); Cgl, *Candida glabrata* (35 RAWs); Cgu, *Candida guilliermondii* (80 RAWs); Clu, Candida lusitanea (101 RAWs); Cor, *Candida orthopsilosis* (32 RAWs); Cpa, *Candida parapsilosis* (43 RAWs); Ctr, *Candida tropicalis* (24 RAWs); Dha, *Debaryomyces hansenii* (59 RAWs); Lel, *Lodderomyces elongisporus* (52 RAWs); WO, *Candida albicans* WO-1 (21 RAWs).

The minimum number of RAWs found was 21 for *Candida albicans* WO-1 and the maximum was 101 RAWs for *Candida lusitanea*. Of note, the number of RAWs identified was different between different strains of the same species, as exemplified in Figure 1 for the reference strain *C. albicans* SC5314 (39 RAWs) and *C. albicans* WO-1 (21 RAWs). These could be due to differences in genome sequencing and assembly or reflect a biologic pattern.

After RAW identification, we selected the reference *C. albicans* genome to BLAST the sequences and map them to the corresponding genes. Using the CGD Gene Ontology with these genes as input, we found that the most represented biological processes (over 20%) are nucleic acids metabolism (DNA and RNA), cell cycle, organelle organization, transport and response to stress, drug or chemical. These categories are consistent with targets that may be used for drug development.

Most regions were only present at a single genomic location, except the two examples that are shown and that should be further investigated (Table 1).

Table 1: Examples of RAWs identified in the *Candida albicans* genome that are good candidates for antifungal targets.

RAW	GENE	FUNCTION
CTTCGACGGCA	orf19.4035	GPI-anchored cell surface protein; transcript induced in model of oral candidiasis
	orf19.6975	Ribosome-associated protein; antigenic in mice
GCGTCGACTAT	orf19.3823	Nonessential protein
	orf19.7285	Ortholog(s) have role in mRNA polyadenylation
	orf19.768	Ortholog(s) have role in signal transduction and plasma membrane localization

The first RAW (CTTCGACGGCA) was found in two genes: orf19.4035 is a cell surface protein that is induced during infection and therefore may be a virulence factor; and orf19.6975, a ribosomal protein that also triggers the immune response in mice.

The second RAW (GCGTCGACTAT) was found in three genes, orf19.3823, orf19.7285 and orf19.768. These have unknown functions, but their orthologues participate in gene expression at the level of transcription (mRNA polyadenylation) or are located in the cell membrane.

In either case, RAWs are present in genes that are usually targeted for drug design [11], such as membrane proteins or components of the transcription and translation apparatus. The fact that each RAW is present in multiple genes also maximizes the potential for the development of antifungal drugs that target several pathways simultaneously.

4 Conclusions and Future Work

Using the alignment-free method previously developed [7], we identified several genomic regions in yeasts that are absent from the human genome. We showed that these RAWs in *C. albicans* are potential targets in the context of antifungal drug development. The results are promising and can be expanded to additional genomes. For example, further analyses will include the study of the conservation degree of these regions between several *Candida* species and the inclusion of filamentous fungi genomes. Also, laboratory tests may be envisaged using siRNAs (small interfering RNAs) or screening of small molecules in vitro.

5 Acknowledgments

This work was partially funded by FEDER (Programa Operacional Factores de CompetitividadeCOMPETE) and by National Funds through the FCT-Foundation for Science and Technology, in the context of the projects UID/CEC/00127/2013, UID/BIM/04501/2013, POCI-01-0145-FEDER-007628, PTCD/EEI-SII/6608/2014, and the grant SFRH/BPD/111148/2015 to RMS.

References

[1] S. Costa-de-Oliveira, I. M. Miranda, R. M. Silva, A. P. Silva, R. Rocha, A. Amorim, A. G. Rodrigues, C. Pina-Vaz. FKS2 Mutations Associated with Decreased Echinocandin Susceptibility of Candida glabrata following Anidulafungin Therapy. *Antimicrob. Agents Chemother.*, 55:1312–1314, 2011.

[2] E. Ricardo, I. M. Miranda, I. Faria-Ramos, R. M. Silva, A. Rodrigues, C. Pina-Vaz. In vivo and in vitro acquisition of resistance to voriconazole by Candida krusei. *Antimicrob. Agents Chemother.*, 58:4604-4611, 2014.

[3] J. Branco, M. Ola, R. M. Silva, E. Fonseca, N. C. Gomes, C. Martins-Cruz, A. P. Silva, A. Silva-Dias, C. Pina-Vaz, C. Erraught, L. Brennan, A. G. Rodrigues, G. Butler, I. M. Miranda. Impact of ERG3 mutations and expression of ergosterol genes controlled by UPC2 and NDT80 in Candida parapsilosis azole resistance. *Clin. Microbiol. Infect.* 23(8):575.e1-575.e8, 2017.

[4] I. Faria-Ramos, P. R. Tavares, S. Farinha, J. Neves-Maia, I. M. Miranda, R. M. Silva, L. Estevinho, C. Pina-Vaz, A. Rodrigues. Environmental azole fungicide, prochloraz, can induce cross-resistance to medical triazoles in Candida glabrata. *FEMS Yeast Res.*, 14:1119-1123, 2014.

[5] A. P. Silva, I. M. Miranda, A. Guida, J. Synnott, R. Rocha, R. M. Silva, A. Amorim, C. Pina-Vaz, G. Butler, A. G. Rodrigues. Transcriptional Profiling of Azole-Resistant Candida parapsilosis Strains. *Antimicrob. Agents Chemother.*, 55:3546-3556, 2011.

[6] J. Branco, A. P. Silva, R. M. Silva, C. Pina-Vaz, G. Butler, A. G. Rodrigues, A. Silva-Dias, I. M. Miranda. Fluconazole and voriconazole resistance in Candida parapsilosis is conferred by gain-of-function mutations in MRR1 transcription factor. *Antimicrob. Agents Chemother.*, 59(10):6629-6633, 2015.

[7] R. M. Silva, D. Pratas, L. Castro, A. J. Pinho, P. J. S. G. Ferreira. Three minimal sequences found in Ebola virus genomes and absent from human DNA. *Bioinformatics*, 31(15):2421-2425, 2015.

[8] A. Zielezinski, S. Vinga, J. Almeida, W. M. Karlowski. Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*, 18:186, 2017.

[9] A. J. Pinho, P. J. S. G. Ferreira, S. P. Garcia, J. M. O. S. Rodrigues. On finding minimal absent words. *BMC Bioinformatics*, 10:137, 2009.

[10] M. Crochemore, A. Héliou, G. Kucherov, L. Mouchard, S. P. Pissis, Y. Ramusat. Minimal Absent Words in a Sliding Window and Applications to On-Line Pattern Matching. In: Klasing R., Zeitoun M. (eds) Fundamentals of Computation Theory. FCT 2017. *Lecture Notes in Computer Science, vol 10472. Springer, Berlin, Heidelberg*, 2017.

[11] T. P. Salci, M. Negri, A. K. R. Abadio, T. I. E. Svidzinski, É. S. Kioshima. Targeting Candida spp. to develop antifungal agents. *Drug Discov Today*, 23(4):802-814, 2018.

Paper:#40

Action Recognition for American Sign Language

By: Phong, N. Ribeiro, B.

Action Recognition for American Sign Language

Nguyen Huu Phong
phong@dei.uc.pt
Bernardete Ribeiro
bribeiro@dei.uc.pt

CISUC – Department of Informatics Engineering
University of Coimbra, Polo II, Pinhal de Marrocos,
3030–290 Coimbra, Portugal

Abstract

In this research, we present our findings to recognize American Sign Language from series of hand gestures. While most researches in literature focus only on static handshapes, our work target dynamic hand gestures. Since dynamic signs dataset are very few, we collect an initial dataset of 150 videos for 10 signs and an extension of 225 videos for 15 signs. We apply transfer learning models in combination with deep neural networks and background subtraction for videos in different temporal settings. Our primarily results show that we can get an accuracy of 0.86 and 0.71 using DenseNet201, LSTM with video sequence of 12 frames accordingly.

1 Introduction

The use of deep learning (DL) for action recognition in video has been an active research in recent years. However, they often fall into recognition of actions as the whole picture e.g. an actor is running, jogging or walking or activities of a group playing football, tennis so on and so forth. Recognizing a more subtle action can be seen in emotion or a more relevant to our research as hand gesture. Though hand gesture is often be limited by number of actions e.g. moving left and right, pointing or twisting etc. On the contrary, sign language offers a more standard and abundance of vocabularies for actions. Just only in America, there are approximately a half of million people using American Sign Language (ASL).

Research in sign languages mostly focus on static images e.g. numbers or alphabets. For example in ASL alphabets, letters J and Z which are dynamic signs are excluded [1, 3]. Some works explore continuous signs – shown as continuous frames – but vocabularies are just static signs [2]. In our research, we analyze dynamic ASL signs where a sign requires at least two or more shapes.

The structure of this paper is as follows. In the next section, we discuss about datasets gathered for this research and present our framework in Section 3. In Section 4, experiments are demonstrated and results analyzed. Conclusions and future works are addressed in Section 5.

2 Dataset

We collected our dataset via ASL dictionaries and resources on Internet for 10 different signs, particularly, referring to animals. These include bear, bird, cat, elephant, fish, giraffe, horse, lion, monkey and mouse. However, we exclude signs having more than one expression e.g. dog (this sign requires tapping on one's hip and optionally twisting thump and index fingers). Overall, we obtain 15 videos for each sign: 10 for training and 5 for testing. Our dataset presents a diversity of signers including 23 females, 10 males and 7 children. Each of them performs maximum 10 signs, minimum 1 sign (18 signers) and 3.75 signs on average. In each video, we cut frames from starting of an action until it ends, within 1-2 seconds long approximately. Figure 1 shows 2 signs sequences for lion and cat. We later extend the dataset to include 15 signs in Section 4.

3 Framework for Action Recognition

We show our architecture for ASL Action Recognition in Figure 2. In this architecture, videos are extracted into frames at different rates e.g. 2 frames/s, 3 frames/s etc. Since visual contents usually share similar elements and training a descent deep net can take several weeks to months, for these reasons, we reuse trained deep networks to filter frames in our architecture and retrain the last layer for our dataset. At this step, we employ different transfer learning models naming InceptionV3, InceptionRestNetV2, Resnet50, DenseNet201 and VGG16/VGG19 to explore which model is the most suitable. Then these frames are extracted to a fixed set of features accordingly.

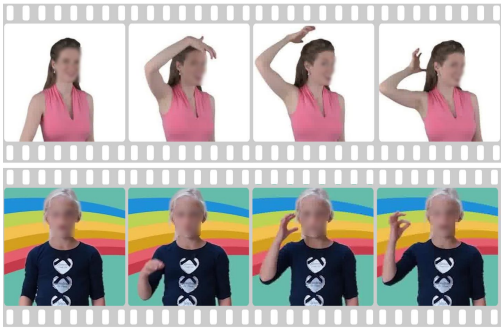


Figure 1: ASL sequences of lion and cat signs

In classifying features from ASL signs, our approach is innovative compared to others in a similar problem endowing classification of only one-shape signs even when signs are in continuous frames. Please note that an one-shape sign can be recognized using only one frame but a dynamic sign requires at least 2 or more frames. We apply Multi-layer Perceptron (MLP) as a baseline on each set of features since the neural network could perform comparable with other more advanced models [4]. Then we perform comparisons of MLP with Long Short Term Memory (LSTM). Performances of these structures are analyzed in next section.

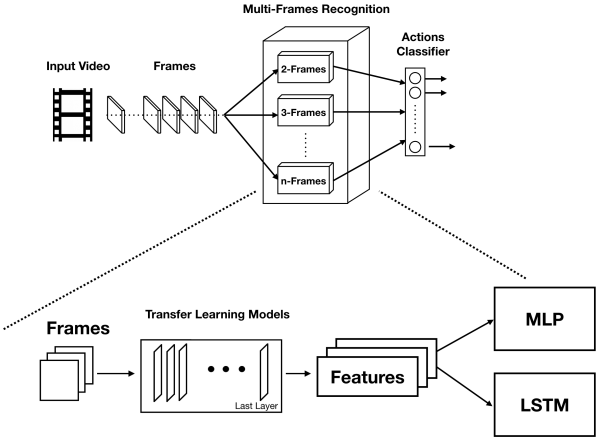


Figure 2: Framework for Action Recognition

4 Experiments & Results

In this section, we fist perform our experiments on ASL raw data for training phase and a pre-processing stage for the testing phase. We vary the length of sequence including 2, 4, 12 and 24 frames per video.

Table 1 shows results for only InceptionResNetV2, InceptionV3 and DenseNet201. The results of ResNes50 and VGG16/VGG19 are excluded from this table since the accuracy is not much better than randomly guess. Performances of MLP and LSTM are also compared. We can observe that MLP usually outperforms LSTM despite of being a simpler structure. This is also interesting to notice that the accuracy of DenseNet201 is better than other models while it is not the finest model on ImageNet dataset. Regarding the choice of sequence length, a sign can be recognized with the highest accuracy using a sequence of just 12 frames. On the other hand, with only the first frame and the middle frame of an ASL sign, an accuracy of approximated 0.8 can be observed.

In Experiment 2, we compare performances of MLP and LSTM on testing data using transfer learning model DenseNet201. From results for

raw ASL data, we can observe that the architecture is not performing well and accuracy are just around 0.3. For this reason, we pre-proceed background subtraction for videos and we also cut the first frame to remove backgrounds left by the subtraction process as we set history window to 1. In addition, the threshold is optimized to 50 since it gives the best accuracy. Beside, we use a median blur filter to remove noise in the frames. We can see from the result that the accuracy is much better with an improved accuracy of 0.58 using LSTM on 12 frames per gesture.

Table 1: ASL data classification on training DL models

	InceptionResNetV2		InceptionV3		DenseNet201	
#Seq Length	MLP	LSTM	MLP	LSTM	MLP	LSTM
2	-	-	0.86	0.80	0.86	0.88
4	-	-	0.97	0.89	0.98	0.89
12	-	-	0.96	0.92	1.00	0.91
24	0.86	0.75	0.93	0.92	0.98	0.94

We find out that the accuracy of 0.58 may not be helpful for many applications and this could not getting better with the current setting. Further investigations point out that actors perform signs differently e.g. actors may repeat a gesture more often than others. For this reason, we strictly reinforce rules for these signs. Likewise, initial signs were replaced for those we could not find enough videos according to the defined rules. After testing several times, we found that the combination of DenseNet201 and LSTM performs the best. Figure 3 and Figure 4 show the accuracy and normalized confusion matrix for the training and testing. We observe the highest accuracy of 0.86 yielded in test. However, in the confusion matrix, we notice that the model misclassifies between a mouse and a cat since their hand positions and shapes are somewhat similar.

In Experiment 4, we extend our dataset from 10 signs to 15 signs including more general conversation terms as finish, hello, love, please and thank you. Confusion matrix is shown in Figure 5. We can see that while several classes e.g. elephant, lion, monkey and tiger are still being recognized correctly, other classes e.g. bear and love are mixed because of their similarity.

Figure 3: Model Accuracy on DenseNet201 after Preprocessing

5 Conclusions and Future works

In our main contribution, we explore the use of several transfer learning models in combination with deep neural networks to classify dynamic ASL signs. We found that an integration of DenseNet201 and LSTM performs the best. In addition, we collected our own database of 150 videos represent 10 signs and an extension of 225 videos that represent 15 signs for verification. We also vary the number of frames per sign to find the best setting.

Our results show that when we vary number of frames per sign, the use of 12 frames gives the highest accuracy. With two frames for one sign, the sign can also be recognized with accuracy of around 0.8. On testing data, to obtain a higher accuracy, we subtract video backgrounds and strictly follow rules for signs. As a consequence, a highest accuracy of 0.86 can be obtained. Our approach different from most of other researches which focus only on static ASL signs and another recent research [5] needs more than one input channel to obtain an accuracy of

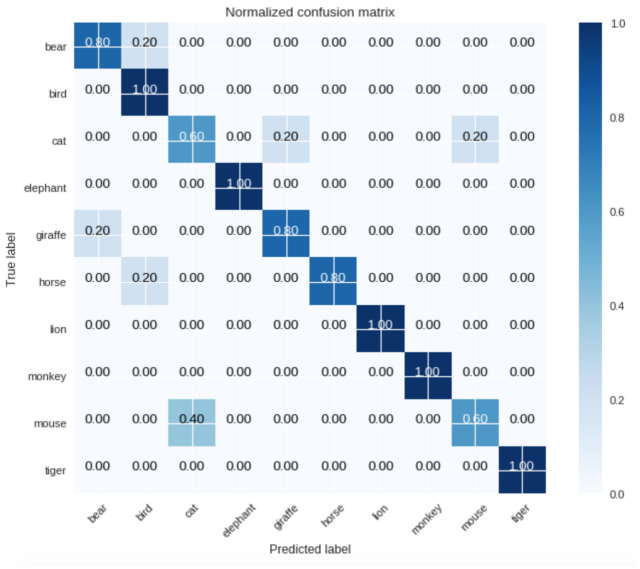


Figure 4: Normalized Confusion Matrix with Accuracy of 0.86 for 10 Signs

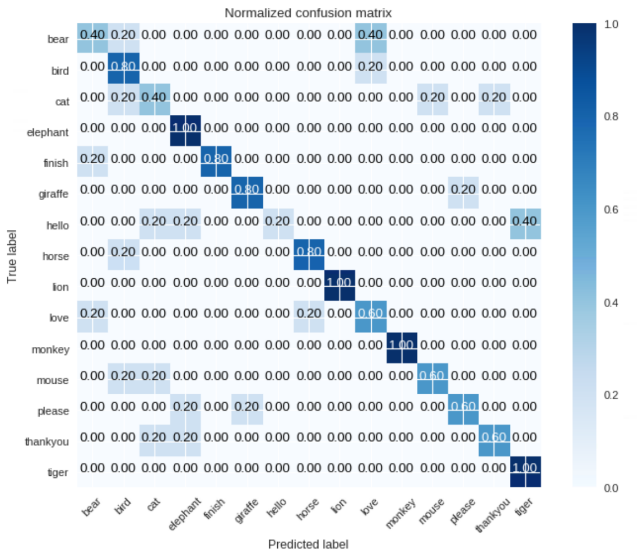


Figure 5: Normalized Confusion Matrix for Extended Signs

0.69. Future work will address extending the dataset and improving our framework to better recognize signs and perform in realtime.

References

[1] Salem Ameen and Sunil Vadera. A convolutional neural network to classify american sign language fingerspelling from depth and colour images. *Expert Systems*, 34(3):e12197, 2017.

[2] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2016.

[3] Alina Kuznetsova, Laura Leal-Taixé, and Bodo Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 83–90, 2013.

[4] Nguyen Huu Phong and Bernardete Ribeiro. Offline and online deep learning for image recognition. In *Experiment@ International Conference (exp. at'17), 2017 4th*, pages 171–175. IEEE, 2017.

[5] Yuancheng Ye, Yingli Tian, Matt Huenerfauth, Jingya Liu, Nataniel Ruiz, Eunji Chong, James M Rehg, Sveinn Palsson, Eirikur Agustsson, Radu Timofte, et al. Recognizing american sign language gestures from within continuous videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2064–2073, 2018.

Paper:#42

SAR missions with commercial UAVs over Mixed-Reality interfaces

By: Rosero, R. Marcillo, D. Grilo, C. Silva, C.

SAR missions with commercial UAVs over Mixed-Reality interfaces

Raul Rosero[1,4]
2170051@my.ipleiria.pt
Diego Marcillo[4]
dmmarcillo@espe.edu.ec
Carlos Grilo[1,2]
carlos.grilo @ipleiria.pt
Catarina Silva[1,3]
catarina@ipleiria.pt

[1]School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2]CIIC, Polytechnic Institute of Leiria, Portugal
[3] Center for Informatics and Systems of the University of Coimbra, Portugal
[4] Universidad de Las Fuerzas Armadas, ESPE, Ecuador

Abstract

The use of unmanned aerial vehicles (UAV) has become an important tool in Search And Rescue (SAR) scenarios, because they are used to preserve human lives and quickly explore areas affected by natural disasters. Also, human body detection has been possible through algorithms which analyse optical and thermal images obtained from the installed cameras.

On the other hand, Ground Stations (GS) with First Person Vision (FPV) interfaces have been implemented with Augmented Reality (AR). Nevertheless, satisfactory projects with commercial UAVs are not common, because these drones are difficult to control for non-expert pilots. These are the reasons why we propose the creation and implementation of an architecture with these requirements. Also, we aim at providing a more user immersive interaction through Mixed Reality (MR) interface over Head-Mounted Display (HMD) glasses.

Keywords: Unmanned Aerial Vehicle, Search and Rescue, Mixed Reality, First Person Vision, Head-mounted Display

1 Introduction

The use of UAVs in SAR operations has protected rescue groups' lives, allowing pilots to be aware of the environment by remotely controlling only the aircraft flight around disaster zones. Injured people location on natural disasters affected areas has been possible thanks to the installation of optical and/or thermal cameras on aerial vehicles and to the transmission of their recorder images in real time to a GS-Ground Station.

In [1] and [2], optical images have been analysed, while in [3], [4], and [5], optical and thermal images have been explored over cross-reference algorithms, allowing the location of human bodies in dark spaces. Nevertheless, [6] mentions that providing operations through First Person Vision interfaces with smart-glasses, GS interfaces of these projects could be more immersive, allowing non-expert users to interact in an easier way.

FPV in smart-glasses provides an interaction of virtual reality with the physical reality throughout the use of networks, sensors, and data bases. On [3], the interaction is called Mixed Reality (Augmented Reality + Virtual Reality) when the pilot sends data to the physical world and does not see the UAV. In [7], the use of commercial UAVs is recommended for disaster management because of their availability, affordability and easy to use.

The target of this work is to implement an architecture that displays an FPV-MR interface for human detection over optical and thermal images in SAR missions, using commercial UAV that provides mission planning, take-off, landing, intelligent flight modes, expertise modes and a Software Development Kit (SDK).

This paper is structured according to discussion points inside the proposed architecture. In Section 2, possible SAR scenarios are described. In Section 3, types, advantages, and disadvantages of commercial UAVs are discussed. Then, in Section 4, methods for human body detection and video transmission. Next, in Section 5, devices for MR with their software provided for developers are presented. In Section 6, the prototype architecture is presented and, finally, future work for the project is delineated.

2 SAR Scenarios

SAR operations must be conducted quickly and efficiently during the first 72 first hours after the disaster hits [7], when injured people have more probability to resist extreme conditions. Geophysical, hydrological, climatological, hydrographical or human-induced disasters could be managed satisfactorily with AUVs because climate conditions have low interferences on the communication with the Ground Station.

In [2], a rescue scenario is described as two parts of a big problem. The first one mentions the identification and injured people location and the second one consists in delivering supplies or resources, but only the first problem is possible due to the physic characteristics of commercial AUVs.

There are two types of SAR operations, Urban SAR and Mountain Rescue [8]. Depending on the features of UAVs, one or both types could be possible. Also, the number of aerial vehicles is an important topic in this operation, because UAVs can organize in teams. Though, [7] mentions that the use of multiple drones in SAR missions does not ensures a satisfactory task.

3 Unmanned Aerial Vehicles

UAVs, also called drones, come in different sizes, from microdrones to large military UAVs [8]. Flight type (rotary-wing or fixed-wing impact) also affects drones' performance in SAR missions. Fixed-wing UAVs move quickly, being ideal for surveying areas and structural inspections, while rotary-wing (helicopters and multi-copters) can do more tedious tasks such as detailed aerial inspection, supply delivery, photography, and filmography.

In [1], [2], and [9] non-commercial helicopters were piloted by experts and had successful SAR missions, while in [4] and [6] commercial quadcopters were easier piloted, because these provide characteristics such as mission planning, take-off, landing and gimbal movements (roll, pitch). Table 1 shows the advantages and disadvantages of different types of UAVs.

UAV Type	Advantages	Disadvantages	
Fixed-wing	Large area coverage	Inconvenient launch and landing	Price
Helicopter	Hover flight Single rotor	Harder to flight than Multicopter	
Multicopter	Availability price Hover flight	Low payload Short flight duration	

Table 1: Comparison of Unmanned Aerial Vehicles for Search and Rescue operations.

4 Architecture prototype

Figure 1 illustrates a prototype of the SAR architecture application to be developed along the project, with some features described in the previous sections. Here, commercial drones can connect with an application throughout an SDK and a "Connection" module. This module connects with the "Video Encoder" module and the interpreter central server, which implements the "Human Detection" with one "Interpreter" module. In the FPV-MR interface, a MR application implements a "Media Receptor" and a "Remote Controller" module in order to control the drone at a distance.

4.1 Real Time Data Sharing Definition

To implement an energy and battery saving mechanism, the UAV application defines a Data Sharing Definition in the "Connection Module". Data, such as a response to remote control events, is sent from the "Connection Module" to the server or it is shared at time intervals. Also, the "Interpreter Module" on the central Server and the "Remote

Controller Module" on the MR application transport this data with the same concept.

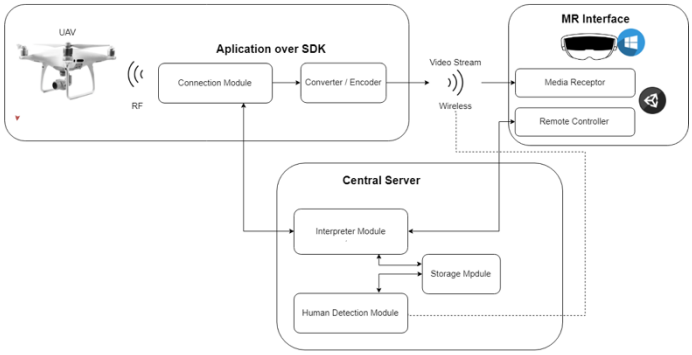


Figure 1: Proposed architecture prototype.

System state, battery state and connection state are data sent at short time intervals so that the drone's health can be controlled. GPS position, altitude, distance, actual position and distance travelled are also sent at short time intervals according the type of SAR mission. On the other hand, state action events, such as a response to remote controller events, are sent from the drone to the remote controller through the Central Server. These data correspond to the drone's configuration or action events and is sent at first instance from the "Remote Controller" such as: flight distance permitted, origin position, GPS mission coordinates, pilot experience, mission type, take-off, landing, copter movements, and gimbal camera movements.

4.2 Data Persistence

Information corresponding to configurations such as: flight distance permitted, origin position, GPS mission coordinates, pilot experience and mission type need to be saved in storage through the "Interpreter Module" and the "Storage Module", to save state configurations of the applications and mission planning configurations. Also, data corresponding to the human detection results will be saved to emit mission reports. This, task is made through the connection of the "Storage Module" with the "Human Detection Module" and the Live Video.

4.3 Human Detection over Live Video

For Live Video with image analysis, this project proposes the use of a similar model to [6], because it defines an architecture to provide video frames to the Ground Station, and the video is passed from the encoder (UAV) to the computing module (GS) where the Computer Vision (CV) works.

Classic human detection algorithms can be implemented together over optical images and thermal image to detect humans in bad weather conditions [4]. Or, a better performance could be obtained with the use of the open source library OpenCV, since it contains more than 2500 optimized algorithms for object detection and needs only to use pre-trained classic algorithms to detect specific objects. In this work, we consider the use of the second approach, with the Histogram of Oriented Gradients HOG for optical images and Haar Cascade based feature for thermal images, as in [4] and [11].

5 Mixed Reality Implementation

The difference between AR and VR can be confused. AR puts digital information in the real world such as HoloLens [10] devices, while VR creates a digital context that simulates the real world such as Oculus Rift and also HoloLens. Nevertheless, as MR systems are sensor-driven, the software architecture is based on data flow.

Mixed Reality glasses as FPV Ground Stations for amateur pilots or video-gamers could provide them with a similar experience to video games or simulator of SAR applications, such as in [8], making tasks easier to complete.

UAVs need to be controlled when pilots cannot see them. For this reason, display control tools are necessary inside MR interfaces. Nowadays, more than one model of glasses with these features can be acquired because Microsoft, among others, such as Dell, HP, Asus, Samsung and Lenovo also developed accessible Head-mounted Displays (HMD) over Windows 10 operating system. In order to make MR interfaces over that, the game developing platform Unity is required, as illustrated in Figure 2, as well as an implementation of a central server interpreter between the drone and the glasses.

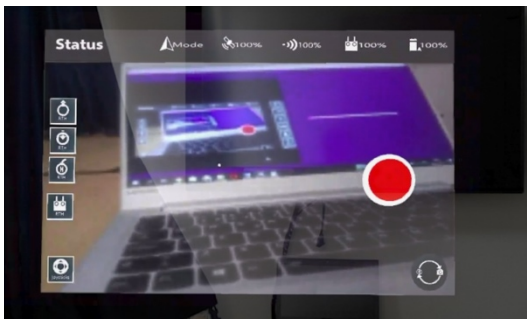


Figure 2: Remote controller prototype in Unity 2017

6 Conclusions and Future Work

In this work we present an architecture and initial implementation of a system for support of SAR missions with commercial UAVs over Mixed-Reality interfaces.

Improvements to the architecture prototype and its implementation will be made according to the limitations of commercial multi-copter drones, which provide the minimal features described before. Next, the modules of the HMD application need to be defined to use the most possible MR glasses models. These modules provide MR interfaces with Unity over Windows 10.

On the other hand, modules of the central server will be adjusted, integrating the human recognition over thermal and optical images transmitted and displayed in the MR interface in real time. Finally, the main goal of this project is to present a generic architecture, that is, agnostic regarding the commercial UAV or the smart-glasses model, which could be used in SAR systems.

References

[1] P. Molina et al., "Searching lost people with UAVs: The system and results of the Close-Search project", Infoscience, Melbourne, 2012.

[2] P. Doherty, R. Rudol, "A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization", M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 1–13, Springer, 2007.

[3] A. Birk et al., "Safety, Security, and Rescue Missions with an Unmanned Aerial Vehicle (UAV)", Journal of Intelligent & Robotic Systems, 64(1), pp 57–76, Springer, 2011.

[4] M. S. Rapee Krerngkamjornkit, "Human Body Detection in Search and Rescue Operation Conducted by Unmanned Aerial Vehicles", Trans Tech Publications, Switzerland, 2013.

[5] R. Rudol, P. Doherty, "Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery", IEEE Aerospace Conference, 2008.

[6] X. Ji, X. Xiang, T. Hu, "Data-driven Augmented Reality Display and Operations for UAV Ground Stations", IEEE 6th Data Driven Control and Learning Systems (DDCLS), 2017.

[7] M. Erdelj et al., "Help from the Sky: Leveraging UAVs for Disaster Management", IEEE Pervasive Computing, 16(1), pp. 24-32, 2017.

[8] A. Khalaf et al., "An Architecture for Simulating Drones in Mixed Reality Games to Explore Future Search and Rescue Scenarios", *15th ISCRAM Conference*, 2018.

[9] J. Neto et al., "A Surveillance Task for a UAV in a Natural Disaster Scenario", IEEE International Symposium on Industrial Electronics, 2012.

[10] Microsoft, "Microsoft HoloLens", Microsoft, 2018. [Online]. Available: https://www.microsoft.com/en-us/hololens. [Accessed 9 9 2018].

[11] R. Ribeiro, J. M. Fernandes and A. Neves, "Face Detection on Infrared Thermal Image", University of Aveiro, Aveiro, 2016.

Paper:#46

Traffic sign recognition using shallow learning techniques

By: Pessoa, D. Lopes, F. Valente, F. Medeiros, J. Teixeira, C.

Traffic sign recognition using shallow learning techniques

Diogo Pessoa
dpessoa@student.dei.uc.pt
Fábio Lopes
fadcl@student.dei.uc.pt
Francisco Valente
pfcv@student.dei.uc.pt
Júlio Medeiros
juliomedeiros@student.dei.uc.pt
César Teixeira
cteixei@dei.uc.pt

Centre for Informatics and Systems,
Department of Informatics Engineering
University of Coimbra
Coimbra, Portugal

Abstract

Traffic Sign Recognition Systems are widely used in the new high technology vehicles, providing an easier and safer driving experience. So, it is extremely necessary to develop pattern recognition systems that are able to identify correctly all the existent traffic signs in the fastest way possible. In this work, we evaluate the use of shallow machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbor (k-NN), Naïve Bayes and Minimum Distance Classifier (MDC), for automatic classification of 43 different traffic signs. Results reveal that k-NN classifier achieve the best accuracy of 95,9%, which approaches the performance obtained with deep convolutional neural networks (CNN), with the advantage of a computationally lighter training.

1 Introduction

Automation has gained an increasingly central role in our daily lives allowing the development of several mechanisms that make life easier for people. In this context, one of the key areas is the autonomous vehicles one. Even though these cars are becoming really capable of perform the needed tasks they still face many challenges such as recognizing perfectly the traffic signs.

One of the difficulties is the speed of the vehicles. As the newest cars are able to reach high speed levels, it is really fundamental for this type of recognition system to have the ability to classify signs in a short interval of time. Furthermore, not only the speed is a problem but also the weather conditions and time of the day as well. During the sunrise and the sunset, the sun is low in the sky, and because of that the system can be confused due to the noise captured by the sensors. Also, in the raining days, the lights of other cars as well as the rain causes some noise to the system.

Even in perfect conditions, though the traffic signals are quite distinguishable to the human eye, at machine learning level there are still some signs that overlap each other since they have similar shapes or colors which makes really important to have a lot of information that provides variety within each class and between all of them. Thus, an analysis of methodologies is needed to understand the difficulties and for further improvement in this field.

2 Related Work

Several works have been done in this field of machine learning area. The competition "Machine Learning for Traffic Sign Recognition" [11] contributed with a great improvement to this area, as participants had to build a classifier that could classify with high accuracy all the traffic signs.

In [1] several deep neural networks architectures to make a multi-column Deep Neural Network (DNN) were considered. The training data that they used was the central ROI of each sample removing the margin which reduces each image to 48x48 pixels. Their algorithm reached an accuracy of 99.46%. In [9], the autors used multi-scale Convolutional Neural Network (CNN). They build the classifier using raw images with just 32x32 pixels. The features were extracted from images during the training phase. This classifier showed an accuracy of 98.31%. A third algorithm was a classifier based on random forest which was simpler than the previous ones [13]. It was build using 500 trees and with Histogram of Oriented Gradients (HOG) features, reaching an accuracy of 96.14%. Manual classification was also considered in order to compare with the automatic one. It was mentioned in [11] that the best human classifier had an accuracy of 99.22% and the average classification accuracy was 98.84%, which are under the best results obtained with the best automatic algorithms.

3 Dataset

The data used in this project was obtained by the *German Institut fur Neuroinformatik* (INI) and contains images of more than 50000 traffic signs. The dataset is called "The German Traffic Sign Benchmark" (GTSRB) and is available at `http://benchmark.ini.rub.de/?section=gtsrb&subsection=news` [12]. This dataset is composed by a training and a testing sub-sets with 39209 and 12630 samples (i.e. images), respectively. In addition to the original images, dor each one, there is included 5 different types of features. All the features were previously extracted and can be grouped in: 3 types of HOG, Haar features and Color Histograms (HueHist).

4 Methods

The work pipeline is composed by the phases: preprocessing; training, validation and selection; and testing. In the testing phase the best selected classifiers were applied to the testing sub-set and the ability to correctly classify new traffic signs samples was evaluated.

4.1 Preprocessing

In the first part, a downsampling [7] technique was performed to the training data due to the non-balanced nature of the data set. Then features engineering techniques were used to better prepare the data for classification. First, we applied a feature selection method and then a feature reduction one.

As feature selection we considered three different approaches, the Kruskal-Wallis (KW) [6] non-parametric statistical analysis, correlation between each feature and their labels in order to find which features had the best relation with the targets and also a combination of both. The best parameters for this methods were retrieved through a grid search and cross validation using a Minimum Distance Classifier due to its simplicity.

Besides that, Principal Component Analysis (PCA) [10] and Linear Discriminant Analysis (LDA) [8] were used as features reduction techniques after feature selection. In PCA the number of dimensions for features reduction was obtained using the Kaiser criterion and in LDA we considered the maximum dimension possible, i.e. 42 dimensions.

4.2 Training, validation and selection

In the second part, a classification of the traffic signs was done using the data that resulted from features selection and reduction process. Several classifiers types were tested: Minimum Distance Classifiers (MDC) - Euclidean and Mahalanobis [4], Support Vector Machine (SVM) - Linear, RBF and Polynomial [2], K-Nearest Neighbors (k-NN) [3] and Naïve Bayes [5].

The best parameters of classifiers were gathered using a grid search and best values were used. For each classifier a 10-fold cross validation

was performed and its accuracy was calculated. At the end the best classifiers from the different types were selected for testing.

4.3 Testing

In the last part, the classifiers with high accuracy in the "Training, validation and selection" phase were used in the testing subset as well as all the approach used before, i.e. best combination of feature selection and reduction. The accuracy of those classifiers was determined in order to evaluate their ability to classify correctly the 43 different traffic signs.

5 Results and Discussion

The Kruskal-Wallis method show better results for feature selection while for feature reduction the LDA technique performs better than the PCA one. The second part was expected because LDA is a supervised approach, so if the data show a good relation with its classes, LDA can achieve a better data separation.

All the classifiers in the validation phase (10-fold cross validation) showed a very high accuracy, over 98.8%. So, we decided to apply all of them in the testing data to verify which one performs best to new traffic signs images. In order to achieve maximum performance, the best parameters were searched in the classifiers which allowed some kind of parameterization. The classifiers that were optimized are as follows: k-NN classifier, the number of neighbours was tested between 1 and 31; linear SVM, the value o C was tested between $C = 2^{-5}$ and $C = 2^{5}$; RBF SVM, the value o C was tested between $C = 2^{-5}$ and $C = 2^{5}$ and the value of Gamma between $\gamma = 2^{-5}$ and $\gamma = 2^{1}$; polynomial SVM, the value o C was tested between $C = 2^{-5}$ and $C = 2^{5}$ and the value of the polinomial order between 1 and 3.

The accuracy obtained for each classifier type in the testing phase is presented in table 1.

Machine Learning Algorithm	Accuracy	Parameters
MDC (Euclidean)	94.7%	N.A.
MDC (Mahalanobis)	94.9%	N.A.
Naive Bayes	93.1%	N.A.
SVM Linear	95.4%	$C = 2^{-4}$
SVM Polynomial	94.8%	$C = 2^{-5}$, Order=3
SVM RBF	95.1%	$C = 2^{-4}$, $\gamma = 0.5$
k-NN	95.9%	K=9

Table 1: Accuracy values.

Table 1 show that the k-NN algorithm is the one that performs better in the testing group. However, the other tested algorithms have pretty high accuracies as well. This was already expected due to the highly number of training samples and highly number of available features providing us all the details about all the signs.

Using confusing matrices we found the following main confusions between: speed limits traffic signs, right hand curve and traffic signal ahead sign, traffic signal ahead and the danger sign, roundabout sign and give away sign. The first three confusions were already expected. In the first one all the signs are round, white with a number in the center, so the only way to distinguish them was by the number or by the strip that represent the end of speed limit, which is not that easy. In the second and third ones, the three traffic signs have the same format, only the symbol in the center is different. The last confusion was not expected however once these traffic signs have a different format and a different color. Nonetheless, the SVM classifiers are the only ones that seemed to overcome this confusion.

6 Conclusion

This study evaluated how well some popular shallow machine learning algorithms as Minimum Distance Classifiers, Support Vector Machines, Naive Bayes and K-Nearest Neighbors classify the different types of traffic signs. Also, this work makes us aware that with an excellent dataset, i.e. a dataset with a great variability of features and samples, that provides a lot of information within each class and between them, it is easier to approach good results with simple classifiers, such as the lazy learner classifier K-Nearest Neighbor. These type of algorithms approach the performance of complex deep machine learning techniques (CNNs and DNNs) and they are much easier and fast to train due to the few parameters that they require.

Lastly, this work shows that we are getting close to a automated world, where the vehicles will be able to make their own choices about the traffic.

References

[1] Dan C. Ciresan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. A committee of neural networks for traffic sign classification. In *IJCNN*, pages 1918–1921. IEEE, 2011. ISBN 978-1-4244-9635-8. URL http://dblp.uni-trier.de/db/conf/ijcnn/ijcnn2011.html#CiresanMMS11.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL https://doi.org/10.1007/BF00994018.

[3] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[4] JP Marques De Sa. *Pattern recognition: concepts, methods and applications*. Springer Science & Business Media, 2012.

[5] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2):103–130, Nov 1997. ISSN 1573-0565. doi: 10.1023/A:1007413511361. URL https://doi.org/10.1023/A:1007413511361.

[6] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. doi: 10.1080/01621459.1952.10483441. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441.

[7] R. Longadge and S. Dongre. Class Imbalance Problem in Data Mining Review. *ArXiv e-prints*, May 2013.

[8] C. Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948. ISSN 00359246. URL http://www.jstor.org/stable/2983775.

[9] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *The 2011 International Joint Conference on Neural Networks*, pages 2809–2813, July 2011. doi: 10.1109/IJCNN.2011.6033589.

[10] J. Shlens. A Tutorial on Principal Component Analysis. *ArXiv e-prints*, April 2014.

[11] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.02.016. URL http://www.sciencedirect.com/science/article/pii/S0893608012000457.

[12] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *In International Joint Conference on Neural Networks*, pages 1453–1460, 2011.

[13] F. Zaklouta, B. Stanciulescu, and O. Hamdoun. Traffic sign classification using k-d trees and random forests. In *The 2011 International Joint Conference on Neural Networks*, pages 2151–2155, July 2011. doi: 10.1109/IJCNN.2011.6033494.

Poster Session 3

Computer Vision, Forecast, Social and Music Applications

Paper:#1

Surface Cameras from Shearing for Disparity Estimation on a Lightfield

By: Monteiro, N. Barreto, J. Gaspar, J.

Surface Cameras from Shearing for Disparity Estimation on a Lightfield

Nuno Barroso Monteiro[1]
nmonteiro@isr.tecnico.ulisboa.pt
João Pedro Barreto[2]
jpbar@isr.uc.pt
José Gaspar[1]
jag@isr.tecnico.ulisboa.pt

[1] Institute for Systems and Robotics
University of Lisbon
Portugal

[2] Institute for Systems and Robotics
University of Coimbra
Portugal

Abstract

Disparity estimation from lightfields is usually based on multi-view stereo geometry, epipolar plane image geometry, or on testing some disparity hypotheses using shearing. Recently, the concept of surface camera image has been used to improve disparity estimation. In this work, we introduce the idea of considering a surface camera image as a generalization of shearing and evaluate the capabilities of using surface camera images in disparity estimation.

1 Introduction

Plenoptic cameras are capable of discriminating the contribution of each light ray emanating from a particular point. The collection of rays captured by these cameras is called a lightfield [8].

In a plenoptic camera, a point in the object space is projected into multiple points in the image sensor. The multiple projections allow to recover disparity assuming no particular position for the cameras, *e.g.* using multi-view stereo [1], or assuming the cameras define a linear path, *e.g.* using the epipolar plane image (EPI) geometry [6]. For the EPI analysis, one can consider gradient based approaches using standard image gradient operators [4] or structure tensors [12]. Nonetheless, these approaches limit the disparity range that can be estimated accurately to one pixel [6]. Shearing of the lightfield [6] increase the disparity range while maintaining the gradient operators constant.

Other strategies test predefined disparity hypothesis by shearing the lightfield and evaluating correspondence and defocus cues on the resulting lightfield [11]. These methods assume lambertian surfaces free of occlusion. Recently, the concept of surface camera images (SCams) [3] has been introduced to identify types of surfaces (lambertian or specular) and occlusions that allow to adapt the metrics used to evaluate correspondences. Although shearing and SCams have been presented as alternative methods, we show that these methods are related and that the SCam is a generalization of the shearing operation.

2 Surface Camera Images as a Generalization

A SCam is a camera that collects rays that intersect at an arbitrary point in the object space [13]. These rays can emanate from different points if the camera's projection center is located on free space (camera A of Figure 2) or is located on a surface point which is partially occluded (camera B of Figure 2). On the other hand, the rays emanate from a common point if the projection center of the camera is defined on a surface point (camera C of Figure 2).

Considering the lightfield in the object space $L_{\Pi}(s,t,u,v)$ acquired by a plenoptic camera with the plane Ω in focus (Figure 1), one can obtain a SCam with projection center at point (q,r) of plane Γ. $L_{\Pi}(s,t,u,v)$ collects rays $\tilde{\mathbf{\Psi}} = [s,t,u,v,1]^T$ that are parameterized using a point (s,t) and a direction (u,v) defined on a plane Π in metric units [10]. Assuming that the plane Γ is at a distance d from the plane Π, one can re-parameterize the lightfield captured by the plenoptic camera relatively to the plane Γ [2], $L_{\Gamma}(q,r,u,v)$, by

$$\tilde{\mathbf{\Psi}}_{\Gamma} = \mathbf{D}\,\tilde{\mathbf{\Psi}} \quad , \tag{1}$$

where

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & d & 0 & 0 \\ 0 & 1 & 0 & d & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad , \tag{2}$$

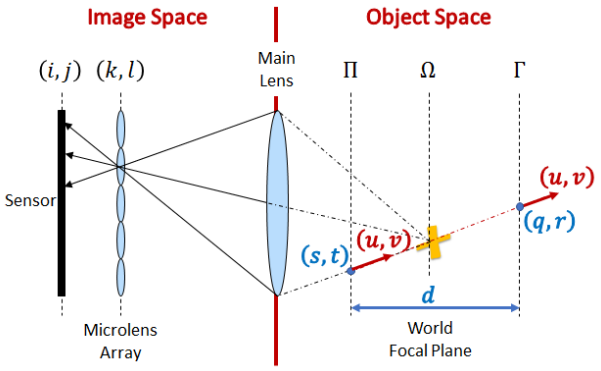


Figure 1: Geometry of a plenoptic camera. On the left, the lightfield in the image space is parameterized using pixels and microlenses indexes. On the right, the lightfields in the object space are parameterized using a point and a direction, and can be parameterized on an arbitrary plane regardless of the original plane Ω in focus.

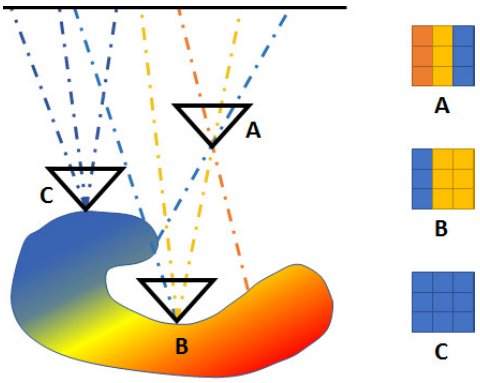


Figure 2: SCams considering different intersection points for the captured rays. The projection center of Camera A does not correspond to a surface point and, therefore, the camera collects rays that emanate from arbitrary surface points. Camera B collects rays that emanate from different points of the surface due to occlusion. Camera C collects rays that emanate from the same surface point. Adapted from Yu *et al.* [13].

and $\tilde{\mathbf{\Psi}}_{\Gamma} = [q,r,u,v,1]^T$ correspond to rays parameterized by a point (q,r) and a direction (u,v) on plane Γ. Notice that the directions remain unchanged with the re-parameterization. Mapping the lightfield in the object space $L_{\Pi}(s,t,u,v)$ to the lightfield in the image space $L(i,j,k,l)$ by the intrinsic matrix $\mathbf{H}$ introduced by Dansereau *et al.* [5], one obtains

$$\tilde{\mathbf{\Psi}}_{\Gamma} = \mathbf{D}\,\mathbf{H}\,\tilde{\mathbf{\Phi}} \quad , \tag{3}$$

with

$$\mathbf{H} = \begin{bmatrix} h_{si} & 0 & h_{sk} & 0 & h_s \\ 0 & h_{tj} & 0 & h_{tl} & h_t \\ h_{ui} & 0 & h_{uk} & 0 & h_u \\ 0 & h_{vj} & 0 & h_{vl} & h_v \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad , \tag{4}$$

where $\tilde{\mathbf{\Phi}} = [i,j,k,l,1]^T$ correspond to rays that are parameterized by pixels and microlenses indexes. The new intrinsic matrix $\mathbf{H}_{\Gamma} = \mathbf{D}\,\mathbf{H}$ allows to relate the lightfield in the object space $L_{\Gamma}(q,r,u,v)$ and the lightfield

in the image space $L(i,j,k,l)$. The mapping between the lightfields allows to define a constraint to identify the rays that intersect at an arbitrary point of the plane Γ. Let $\mathbf{\Phi}_a$ and $\mathbf{\Phi}_b$ be two rays with the same coordinates (q,r) on plane Γ, by taking their difference one defines a constraint on the lightfield coordinates to define a SCam

$$\begin{bmatrix}0\\0\end{bmatrix} = \mathbf{H}_{ij}^{qr}\begin{bmatrix}i-i_r\\j-j_r\end{bmatrix} + \mathbf{H}_{kl}^{qr}\begin{bmatrix}k-k_r\\l-l_r\end{bmatrix} \quad , \tag{5}$$

where (i_r,j_r,k_r,l_r) are reference coordinates to enforce the constraint, and $\mathbf{H}_{(\cdot)}^{qr}$ corresponds to 2×2 sub-matrices of $\mathbf{H}_\Gamma$ obtained from selecting the entries of the first two rows, denoted by qr, and selecting either the entries of the 1^{st} and 2^{nd} columns, denoted by ij, or the 3^{rd} and 4^{th} columns, denoted by kl. Using the constraint (5) and assuming that we want to maximize the number of rays (see Section 4.1 of [10]) that define a SCam, one obtains a sampling on (i,j) for disparities lower or equal than one

$$k = k_r + \beta_{ik}\,(i-i_r) \wedge l = l_r + \beta_{jl}\,(j-j_r) \quad , \tag{6}$$

and a sampling on (k,l) for disparities greater than one

$$i = i_r + \beta_{ik}^{-1}\,(k-k_r) \wedge j = j_r + \beta_{jl}^{-1}\,(l-l_r) \quad . \tag{7}$$

The parameters $\beta_{ik} = -\frac{h_{vi}+d\ h_{ui}}{h_{sk}+d\ h_{uk}}$ and $\beta_{jl} = -\frac{h_{tj}+d\ h_{vj}}{h_{tl}+d\ h_{vl}}$ correspond to the disparities considered on the viewpoint images [1] for a point at depth d [10]. The sampling defined in equation (6) correspond to the sampling performed during the shearing operation defined by Tao *et al.* [11] considering $\beta_{ik} = \beta_{jl} = 1 - \frac{1}{\alpha}$.

The SCam defines a camera with an arbitrary position for the projection center but one is interested on cameras whose projection centers lie on the scene surfaces. These cameras collect rays that originate at the same surface point. On the other hand, in shearing, one wants to shear the EPIs in order to have the rays that originate at a given surface point in the same microlens, i.e. the information present in the microlenses of the sheared lightfield correspond to the SCams. Furthermore, the SCams are a generalization of the shearing operation on the lightfield by considering the sampling on (k,l) for disparities greater than one. Remember that shearing considers the sampling on (i,j) regardless of the disparity being evaluated.

3 Results

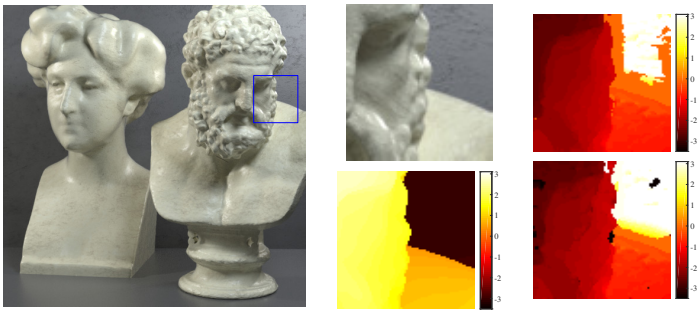
In this section, we compare the disparity maps obtained from evaluating different correspondence cues for the Greek dataset of the 4D Lightfield Benchmark [7]. The dataset is not fully analyzed, but instead a small region with 80.56% of pixels with absolute disparity values greater than one. The dataset is illustrated in Figure 3.a-b. Notice that a dense disparity map is obtained considering a disparity estimation framework that comprises several steps like filtering and refining the disparity cost volume [11] or a densification strategy like Total Variation regularization [9]. Nonetheless, in this work, we only evaluate the quality of the initial disparity map.

The disparity maps are obtained by testing different disparity hypothesis and evaluating the metrics that define the correspondence cues [3, 11] on the sheared lightfield and on the SCams. The occlusion is handled using the strategy defined in Chen *et al.* [3]. The results are exhibited on Figure 3.c, and the disparity errors are summarized in Table 1. The errors in this table discard pixels in homogeneous regions. Table 1 shows that the analysis using SCams provides more accurate results for pixels with ground truth disparity greater than one. For example, using the correspondence cue [3] (denoted as CNS), the disparity estimation improves by 64.19% for pixels with disparity greater than one.

Cues	Shearing Disparity Error		SCams Disparity Error	
	$\lvert Disparity\rvert > 1$	$\lvert Disparity\rvert \leq 1$	$\lvert Disparity\rvert > 1$	$\lvert Disparity\rvert \leq 1$
Correspondence [3, 11]	3.9819	1.4138	3.9811	1.4322
Correspondence CNS [3]	5.2353	1.7707	3.3607	3.2936

Table 1: Disparity errors obtained for the Greek dataset of the 4D Lightfield Benchmark [7] by evaluating correspondence cues [3, 11].

[1] A viewpoint or sub-aperture image is obtained by selecting and combining the rays that reach the same pixel of each microlens, i.e, by selecting the pixel (i,j) of each microlens (k,l).



(a) Central Viewpoint (b) ROI and GT (c) Corresp. [3, 11]

Figure 3: Greek dataset [7] used to evaluate the disparity map obtained using shearing and SCams. **(a)** central viewpoint image. **(b)** selected region and corresponding ground truth disparity map. **(c)** disparity maps from correspondence cue [11] applied to a subset of unoccluded rays [3] on the sheared lightfield (first row) and on the SCams (second row).

4 Conclusions

In this work, we defined the constraint that allows to obtain a SCam for a plenoptic camera using the camera model defined by Dansereau *et al.* [5], and showed that a SCam is a generalization of the shearing operation on the lightfield by introducing a sampling on the microlenses coordinates (k,l). The capabilities of the SCams were evaluated on the Greek dataset of the 4D Lightfield Benchmark [7] using different correspondence cues [3, 11]. The results obtained suggest that the SCams are more accurate for pixels with disparity greater than one.

Acknowledgements

Work partially supported by the FCT project UID / EEA / 50009 / 2013.

References

[1] Edward H Adelson and John Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2): 99–106, 1992.

[2] Clemens Birklbauer and Oliver Bimber. Panorama light-field imaging. In *Computer Graphics Forum*, volume 33, pages 43–52. Wiley Online Library, 2014.

[3] Can Chen, Haiting Lin, Zhan Yu, Sing Bing Kang, and Jingyi Yu. Light field stereo matching using bilateral statistics of surface cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1518–1525, 2014.

[4] Don Dansereau and Len Bruton. Gradient-based depth estimation from 4d light fields. In *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, volume 3, pages III–549. IEEE, 2004.

[5] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1027–1034, 2013.

[6] Maximilian Diebold and Bastian Goldluecke. Epipolar plane image refocusing for improved depth estimation and occlusion handling. 2013.

[7] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*, pages 19–34. Springer, 2016.

[8] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[9] Nuno Barroso Monteiro, Joao Pedro Barreto, and José Gaspar. Dense light-field disparity estimation using total variation regularization. In *International Conference Image Analysis and Recognition*, pages 462–469. Springer, 2016.

[10] Nuno Barroso Monteiro, Simão Marto, João Pedro Barreto, and José Gaspar. Depth range accuracy for plenoptic cameras. *Computer Vision and Image Understanding*, 2018.

[11] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 673–680, 2013.

[12] Sven Wanner and Bastian Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(3):606–619, 2014.

[13] Jingyi Yu, Leonard McMillan, and Steven Gortler. Scam light field rendering. In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*, pages 137–144. IEEE, 2002.

Paper:#6

Use of Epipolar Images Towards Outliers Extraction in Depth Images

By: Celorico, D. Cruz, L. Dihl, L. Gonçalves, N.

Use of Epipolar Images Towards Outliers Extraction in Depth Images

Dirce Celorico[1]
dircelorico@isr.uc.pt
Leandro Cruz[1]
lmvcruz@isr.uc.pt
Leandro Dihl[1]
leandro.dihl@isr.uc.pt
Nuno Gonçalves[12]
nunogon@deec.uc.pt

[1] Institute of Systems and Robotics
University of Coimbra
Coimbra, Portugal

[2] Portuguese Mint and
Official Printing Office
Lisboa, Portugal

Abstract

Plenoptic cameras, such as Lytro and Raytrix, has been widely used over the last years. Their main feature is the light intensity acquisition from several viewpoints. From these viewpoint images, we can reconstruct a 3D model of the captured scene by calculating the depth of each pixel by a passive depth estimation, using only one captured image. In this way, the depth denotes the distance between the respective point and the camera.

Although this 3D model can be directly used for several purposes such as refocusing after capture or object segmentation, they are often quite noisy, what is a disadvantage for some applications like 3D visualization, or more complex mesh processing.

In this work, we will present a method for filtering the depth model, reconstructed from light field cameras, based on the removal of low confidence reconstructed values and using an inpainting method to replace them. This approach has shown good results for outliers removal.

1 Introduction

Nowadays, the use of light field (or plenoptic cameras) has been much more common not only because of progress on calibration and decoding but due mainly to depth estimation improvement. These special cameras were introduced by Ng [5]. Their main feature corresponds to the use of an array of microlenses in front of the sensor, which essentially achieves the capture of an array of views simultaneously, in a single shot image, without special patterns projection in the scene.

It allows the user to acquire not only information regarding the intensity of light in the scene but also the information related to the direction of the light rays in the space. This information is gathered from different viewpoints, thus after mathematical manipulation, it allows to extract a 3D model of the scene.

One of the most important parameters obtained by this type of camera is the depth of each point (the distance between the point in the scene and the camera). From a single raw image of a plenoptic camera the depth can be estimated, at least in positions with sufficient local contrast. The ability to determine depth perception and to be able to reconstruct the scene in a 3D environment has become of major importance in tasks such as segmentation, detection or even 3D display.

There are many works about depth estimation [2]. More recently, depth estimation approaches from light-field have handled the epipolar plane images (EPIs) geometry due to this model type has shown robust and correctness results in depth map achievement. In this work, we will keep focus on outliers filtering from a local coherence point removal. After the removal of the points, we fill the produced holes by the usage of an inpainting method.

2 Related Work

For the past few years, the development in the study of light field cameras has provided a range of techniques to accurately estimate the depth. Related works with this kind of cameras are still considered current and has produced such a useful progress as in Monteiro et al. [4]. The use of the epipolar geometry (Epi's) in order to define a depth map is one of the most used techniques, and it presents some better results. For instance, in works like Wanner and Goldluecke [10], Suzuki et al. [9], Lin et al. [6] Si et al. [3] and Zhang et al. [12].

The plenoptic function is represented as a seven-dimensional space (illustrated in equation (1)) and it represents the amount of data in a scene.

$$L = (v_x, v_y, v_z, \phi_x, \phi_y, \lambda, t) \tag{1}$$

It consists of the information of intensity for every 3D point (v_x, v_y, v_z), the corresponding direction (ϕ_x, ϕ_y), wavelength (λ) and time (t). The reduction of variables from 7D to 4D, the so-called Lumigraph, was introduced by Gortler et al. [8]. Taking into account that image acquisition is made by a single shot capture the variable time can be neglected, and it is evenly possible to neglect the variable wavelength by considering a grey scale environment.

We also adopted in this work the two-plane parametrization approach which is widely studied and represents a simple structure where each ray is defined by the intersection with two parallel planes. One corresponding to the camera plane, the other one to the image plane, and both are separated by a distance $z = 1$.

Therefore, a light field can then be represented as a 4D function $f(x,y,s,t)$ where the dimensions (x,y) represent the spatial distribution and the dimensions (s,t) represent the angular distribution. By fixing one of the spatial dimension and one angular dimension we can get a horizontal epipolar plane image S_{t^*,y^*} or a vertical epipolar plane image S_{s^*,x^*}.

Regarding the inpainting approach, it is considered a world-wide known technique to fill lost information in an image, it has been developed accordingly to different algorithms that conduct to the filling of gaps in images [7]. Some of these different approaches rely on Partial Differential Equations (PDE), texture synthesis, exemplar-based search or even wavelet transforms.

3 Depth Estimation

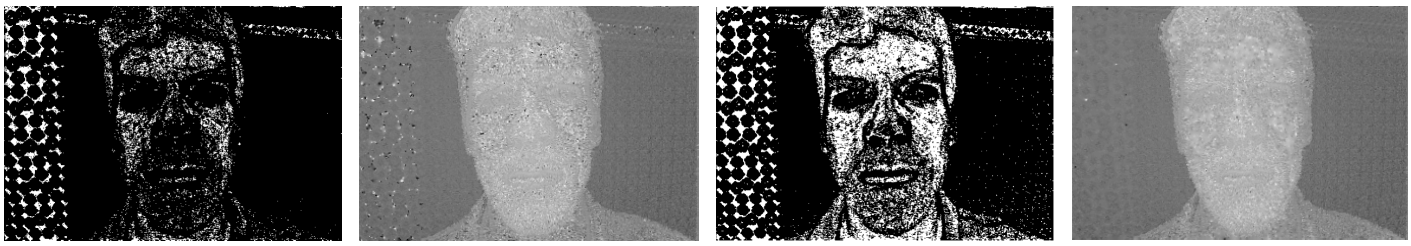
The input of our filtering method is the raw image captured by a light field camera while the output is an RGBD image (the captured colour and the reconstructed depth). From the raw image, we use the epipolar plane images analyses approach so images such as Figure 1 are obtained.

Figure 1: Epipolar plane image example S_{t^*,y^*}.

In an epipolar image, a line corresponds to a point across different viewpoint images. The corresponding slope in the lines can be used to estimate the respective depth value since they are directly related. In order to determine the slope in epipolar images we used the structure tensor, an approach presented by Wanner and Goldluecke [11], and we also defined the reliability assigned to the depth throughout the coherence value. This structure tensor is defined by equation (2). In equation (3) it is represented

(a) (b) (c)

Figure 2: (a) Light field image used; (b) Depth image; (c) Coherence image (with values in the interval [0 1])



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

Figure 3: (a) Mask from values bellow threshold of 0.6; (b) Depth image after inpainting (0.6) ; (c) Mask from values bellow threshold of 0.8 ; (d) Depth image after inpainting (0.8)

the angle to determine the direction of the lines. Finally, equation (4) calculates the reliability parameter. This procedure is implemented for each horizontal (S_{t^*,y^*}) and vertical (S_{s^*,x^*}) plane image and for all viewpoints. For each pixel in the image and depending on the coherence value of the vertical and the horizontal Epi, a selection is performed between them to accomplish a single value for the depth at that pixel.

$$J = \begin{bmatrix} G_\sigma * (S_x S_x) & G_\sigma * (S_x S_y) \\ G_\sigma * (S_x S_y) & G_\sigma * (S_y S_y) \end{bmatrix} = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix} \tag{2}$$

where G represents a Gaussian operator, and S_x and S_y represent the gradients of the Epi in x and y directions respectively.

$$\phi = \frac{1}{2}\arctan\left(\frac{Jyy - Jxx}{2Jxy}\right) \tag{3}$$

$$r_{y^*,t^*} = \frac{\left(J_{yy} - J_{xx}\right)^2 + 4J_{xy}^2}{\left(J_{yy} + J_{xx}\right)^2} \tag{4}$$

An example of one of the image viewpoints is represented in Figure 2(a).

In Figure 2(b) we can see the corresponding depth map achieved with the related coherence image (Figure 2(c)).

In the coherence image we are using a heat colour scheme in order to illustrate the variations of coherence in the image. The red one corresponds to a higher level of reliability (1) and the blue one to a low reliability value (0). As expected, we can notice that the coherence presents lower values in areas with low texture such as the cheeks and forehead.

4 Low Coherence Removal

Following the coherence map creation, we remove all low confidence points. This is performed with the use of a threshold. In This way, coherence values located below the threshold are considered null. The holes created with this removal are filled by using an inpainting method. [1]

This threshold is such an important parameter to achieve superior results. The larger threshold choice the larger the holes, what can imply poor reconstruction. On the other hand, the lower threshold choice the smaller confidence, which implies low noise removal.

For an easier way to illustrate the difference between regions considered for inpainting, we decided to do it with two thresholds values of 0.6 and 0.8. In Figure 3, it is illustrated two masks generated employing these threshold values. For each example, it is also shown the respective depth map result achieved after inpaiting.

In these cases pixels in the depth map that have a coherence value below the threshold are discarded since they represent poorly reconstruction estimation. The second case is a good example in which the threshold trade off was well satisfied so that it was removed most of noisy points, however the resulting holes were not too large, what meets a satisfactory reconstruction.

The inpainting procedure considered [1] solves a Partial Differential Equations (PDE) to fill up the hole by a continuous interpolation of the boundary value (frontier constraint). This method assumes springs (with a nominal length of zero) connect to each node with every neighbour (horizontally, vertically and diagonally). The result is a piece of surface that can be glued to the depth model in a continuous way, obtaining a derivative continuity as well (a smooth surface).

5 Results and Future Work

In this work, we presented a fast outliers filtering method for depth models created using plenoptic cameras. The models obtained by this type of camera often need to be filtered due to the presented high frequency of noise and holes. Furthermore, the proposed filtering can be combined to a low-pass filter to smooth the model (and the outliers removal improve the result of this other filter by reducing the frequency cut).

The proposed method can also be improved in some directions. For instance, in the way we calculate the confidence threshold to be assumed for the inpainting procedure. We have currently been researching a relation between the threshold and the coherence energy (then, how to cut in coherence) in such a way that maximize the coherence average without increasing dramatically the hole size (what can create spurious inpainting reconstruction). To do so, a positive solution might be the usage of a local threshold.

Since this is an ongoing project, one of the following steps is to apply this method to a set of images with depth ground truth. It allows measuring the method accuracy and compare it with other state of the art approaches.

Current results underline the viability of outliers filtering from a local coherence point removal. This method is promising and expands other researches about depth estimation and filtering which are left as future work.

References

[1] J. D'Errico. Inpaint nans. `https://www.mathworks.com/matlabcentral/fileexchange/4551-inpaint_nans`.

[2] A. Jarabo Y. Zhang L. Wang Q. Dai T. Chai G. Wu, B. Masia and Y. Liu. Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):926–954, Oct 2017.

[3] H. Zhu L. Si and Q. Wang. Epipolar plane image rectification and flat surface detection in light field. *Journal of Electrical and Computer Engineering*, 2017, 2017. `https://doi.org/10.1155/2017/6142795`.

[4] J. P. Barreto N. B. Monteiro, S. Marto and J. Gaspar. Depth range accuracy for plenoptic cameras. *Computer Vision and Image Understanding*, 168(C), 2018. ISSN 1077-3142.

[5] R. Ng. Digital light field photography. *Stanford University Ph.D. thesis.*, 2006.

[6] F. Wu P. Lin, J. Yeh and Y. Chuang. Depth estimation for lytro images by adaptive window matching on epi. *Journal of Imaging*, 3(17), 2017.

[7] Mr. Krunal R. Patel R. Suthar. A survey on various image inpainting techniques to restore image. *Int. Journal of Engineering Research and Applications*, 4(2):85–88, 2004.

[8] R. Szeliski S. Gortler, R. Grzeszczuk and M. Cohen. The lumigraph. In *In Proc. ACM SIGGRAPH*, pages 43–54, 1996.

[9] Takahiro Suzuki, Keita Takahashi, and Toshiaki Fujii. Sheared epi analysis for disparity estimation from light fields. *IEICE Transactions on Information and Systems*, E100.D(9):1984–1993, 2017.

[10] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4d light fields. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48, June 2012.

[11] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):606–619, 2014.

[12] Y. Liu H. Wang X. Wang Q. Huang X. Xiang Y. Zhang, H. Lv and Q.i Dai. Light-field depth estimation via epipolar plane image analysis and locally linear embedding. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):606–619, 2017.

Paper:#7

Uniquemark: A computer vision system for hallmarks authentication

By: Barata, R. Cruz, L. Gonçalves, N.

Uniquemark: A Computer Vision System for Hallmarks Authentication

Ricardo Barata[1]
rbarata@isr.uc.pt
Leandro Cruz[1]
lmvcruz@isr.uc.pt
Bruno Patrão[1]
bpatrao@isr.uc.pt
Nuno Gonçalves[12]
nunogon@deec.uc.pt

[1] Institute of Systems and Robotics, University of Coimbra Coimbra, Portugal
[2] Portuguese Mint and Official Printing Office Lisboa, Portugal

Abstract

We are presenting Uniquemark[1], a vision system for authentication based on random marks, particularly hallmarks. Hallmarks are worldwide used to authenticate and attest the legal fineness of precious metal artefacts. Usually, these artefacts are marked with a punch, which embeds on the surface of the metal an illustration (Fig. fig:arq b) and c). In addition to the illustration, we propose the deposition of randomly scattered micro-diamonds ($\approx 50\mu m$)on the metal surface in order to create a unique random pattern. Diamond particles patterns are randomly produced, and the probability of two equal patterns coexist is null. By detecting patterns on a precious metal piece, we can authenticate it. Our authentication method is based on a multiclass classifier model that uses mark descriptor composed by several geometric features of the particles. The proposed authentication system has met better results in both real examples and simulations.

1 Introduction

In this work, we use marks made from scattered particles over a metallic support surface to create random patterns. The main assumption of our authentication proposal is: the probability of the particles spreading process to generate two identical marks is zero. Most pattern recognition systems describe data through image-based features [1, 3]. The most important contribution of this work is the usage of geometric features to depict a unique random scattering of particles, shaping an authentication system. Such mark creation is meant to be integrated to a large-scale production environment like Assay Offices. Marks are randomly produced by scattering diamond particles over the metal surface before the official hallmark application. This procedure of mark production turns its counterfeit nearly impossible.

2 Methodology

Authentication is performed in two steps: registration and identification. The first involves mark description and subsequent training of the classifier. The second also involves the description of a given mark, followed by the search for the most similar mark previously registered and the verification whether it is a true match. Both processes start by acquiring images of marks (Section 2.1). Afterwards, it is performed the mark detection (Section 2.1): (i) region of interest detection and (ii) its rectification and crop. Over the region of interest, the segmentation is applied to identify the particles, then it is defined a respective key point for each of them (Section 2.2). The mark description (Section 2.3) consists of the creation of a vector that characterizes key points scattering. In Section 2.4, it is presented our mark registration method and the identification approach.

2.1 Mark Detection

Once acquired the hallmark image, we crop the respective image tight to the mark edges (Fig. 1 b)). It follows that such step requires the mark to be detected. In this way, mark detection is performed by a U-Net Convolutional Neural Network (CNN) [4] which defines whether a pixel belongs to the mark or not. The network is trained with a set of 132 coloured images and its handcrafted label masks by means of Stochastic Gradient Descent with Momentum method to update the weights, over 150 epochs. Although the images are acquired using a high-resolution device, they are downsampled to 256×256.

[1] Patent Application: INPI 20181000043541

2.2 Particle Detection

Particle detection process determines a key point for each particle in the mark. It consists of a binary classification process, carried out by employing a U-net neural network [4], that defines whether each pixel of the rectified image belongs to a particle or not (Fig. 1 c). As a result, the network output experiences a series of erosion operations until the segmentation mask becames noise and big particle agglomerates free. For each connected component on the mask, we associate a key point to its centre of mass.

2.3 Description

The descriptor indicates a global representation of a geometric feature distribution of the scattered key points. Likewise, it is scale and rotation invariant. It therefore follows that the mark description is based on a histogram of distances between each pair of key points (Fig. 1 c)).

It should be borne in mind that our descriptor is a histogram that contains 142 bins. In this way, we assume that the key points are in a normalized square whose length of each side is 100. Hence, we assess the distance between all key point pairs and approximate the value to the closest integer (the bin index).

It is also important to highlight that this is a global descriptor whose dimension (142×1 vector) does not depend on the input size (amount of key points) so that we can compare different size marks. For instance, if during the detection step some particles are not identified, we are still able to compare this description with the registered one.

2.4 Registration and identification

The registration step (Fig. 1 f)) give rise to the creation of the authentication model using the representation provided by the descriptor, introduced in previous section. The authentication is treated here as a multiclass classification problem [2]. For this reason, each mark is acquired from different poses and lighting conditions, what generates a set of samples that define a class.

With respect to the identification process, it is performed by means of a Nearest Neighbour method. Besides, to speed up the search process, we use a kd-tree [5] for the nearest neighbour search. Additionally, in this process (Fig. 1 (e)) the feature vector of the mark under test is provided to the kd-tree which returns the closest mark registered in the database. Thus, if the distance between the mark descriptor under test and the registered mark is lower than a certain threshold, hallmark authenticity is confirmed.

3 Results and Discussion

3.1 Simulations

Simulations on synthetic data and tests on real databases were applied to assess our authentication process performance. It was conducted tests which targeted several pattern databases, each of them showing different features.

Table 1 exhibit three different simulation batches performed to validate the identification process (Section 3.1). We decided to apply accuracy as the metric to compare results. We also performed tests in real marks to validate the authentication system as a whole (Section 3.2).

Simulations on synthetic databases include patterns with different key points and artificial noise quantities. Doing so, allows simulating flaws

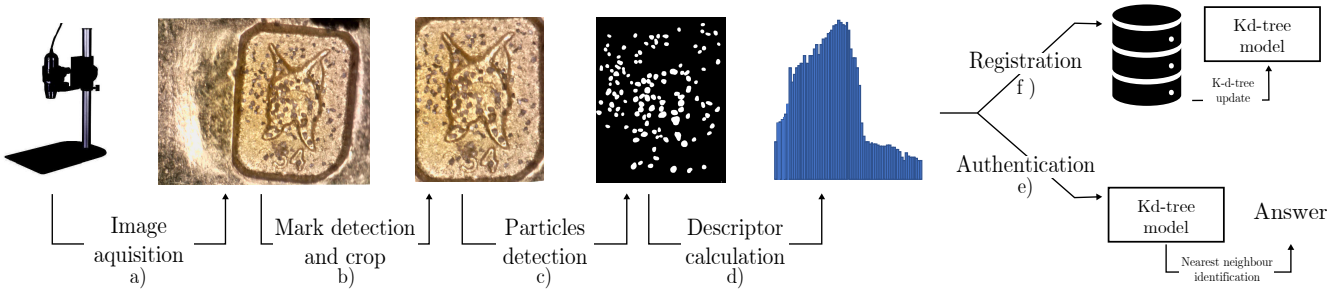


Figure 1: Brief registration and authentication system scheme.

occurred during the particle detection and aging that creates small alterations on marks. Each synthetic pattern was created by randomly scattering a predetermined number of key points on a 100×100 grid cells. Secondly, we added noisy transformations, i.e., the removal, addition and translation of key points to a neighbour cell. The number of removed, added or even translated key points was defined as a percentage of their initial quantity.

For each pattern, we created 14 variations containing different noisy transformations. Among these, 11 were used to train the Kd-tree, and the other three ones were used to validate its performance.

The first simulation was performed for the propose of noise effect assessment (1st batch in Table 1). Consequently, we generated four databases containing 10.000 patterns (and we used a total of 30.000 samples for testing). We also applied different amounts of noise to the databases that had patterns composed by 100 particles. The second simulation (2nd batch) analysed the effect of the number of particles. Finally, the third simulation (3rd batch) evaluated the scalability of our classification model. In this case, each database contained 100.000 patterns.

Table 1: Simulation results.

Number of patterns	Number of key points	Addition, removal and translation noise (%)	Model accuracy (%)
10.000	100	5	100.00
10.000	100	10	99.42
10.000	100	15	95.59
10.000	100	20	84.58
10.000	10	15	99.99
10.000	20	15	98.98
10.000	30	15	99.69
10.000	40	15	98.55
10.000	50	15	98.85
100.000	100	5	100.00
100.000	100	10	99.02
100.000	100	15	92.69
100.000	100	20	74.83

Regarding Table 1, it goes without saying that since the noise increases, accuracy drops. At this point, it should be emphasized that even though the introduced noise is 15%, our model is still able to successfully identify 28.677 of the 30.000 test samples (95.59%).

Furthermore, on the second simulation batch, we kept the amount of noise at 15%, and results call for a reduction of key points number on patterns does not affect method performance.

Regarding the simulations that tested the model capacity to scale towards larger databases, there was no difference between the 10.000 and 100.000 tests when applying 5% of noise (addition/removal and translations). For high levels of noise (10%, 15% and 20%) the accuracy drops 0.40pp, 2.95pp and 9.75pp (percent points), from 10.000 to 100.000 patterns database.

3.2 Real Data Tests

Image acquisition device depends on the purpose. The marks were captured using a medium magnification microscope (Fig. 1 a), and a smartphone with a macro lens.

We collected images from gold and silver samples which were individually tested. From the gold sample it was registered 17 hallmarks, and from the silver sample, 33 hallmarks. From each mark were taken 9 acquisition, 6 used to build the kd-tree, and the remaining 3 were used to testing purposes. The same testing methodology was carried out for the smartphone acquisitions.

The results obtained are presented in the Table 2. As it was expected, the accuracy for the samples acquired with the microscope was higher than the one in the samples acquired with the smartphone. The superior magnification allows a better segmentation of the diamond particles which affect the results downstream. The accuracy is also higher for the marks on gold due to the metal physical features and particle detection is better archived.

Table 2: Results of the real tests

	Gold sample	Silver sample
Microscope	100%	91%
Smartphone	88%	77%

4 Conclusion

In this paper, we presented a vision system for authentication based on random patterns that may be massively produced, Uniquemark[1]. These patterns can be used to identify products, people, etc., since their randomness makes them unique. In this case, we focused on the authentication of precious metal artefacts through official hallmarks.

We have conducted tests on real data and we have taken interesting results. We also have realized that the most delicate step on our system is the detection/segmentation of particles because it highly relies on the type of mark. To sum up, we have performed simulations on synthetic data that does not depend on detection/segmentation, and we obtained high levels of accuracy, that also proves that scalability of the system in the sense of quantity of marks.

References

[1] H. Ali, M. J. E. Salami, and Wahyudi. Iris recognition system by using support vector machines. In *2008 International Conference on Computer and Communication Engineering*, May 2008.

[2] Maya R. Gupta, Samy Bengio, and Jason Weston. Training highly multiclass classifiers. *Journal of Machine Learning Research*, 2014.

[3] Maria De Marsico, Alfredo Petrosino, and Stefano Ricciardi. Iris recognition through machine learning techniques: A survey. *Pattern Recognition Letters*, 2016.

[4] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[5] Robert F. Sproull. Refinements to nearest-neighbor searching ink-dimensional trees. *Algorithmica*, 1991.

[1] Patent Application: INPI 20181000043541

Paper:#8

Graphic Code: Creation, Detection and Recognition

By: Patrão, B. Cruz, L. Gonçalves, N.

Graphic Code: Creation, Detection and Recognition

Leandro Cruz[1]
lmvcruz@isr.uc.pt
Bruno Patrão[1]
bpatrao@isr.uc.pt
Nuno Gonçalves[12]
nunogon@deec.uc.pt

[1] Institute of Systems and Robotics, University of Coimbra Coimbra, Portugal
[2] Portuguese Mint and Official Printing Office Lisboa, Portugal

Abstract

Graphic Code[1] is a new Machine Readable Coding (MRC) method. It creates coded images by organising available primitive graphic units arranged according to some predefined patterns. Some of these patterns are previously associated with symbols used to compose the messages and to define a dictionary. According to the same coding principle presented in this work, we can develop three kinds of graphic codes, each of them able to create a very different coding styles (black and white pixel-based, coloured pixel-based, and icon-based ones). It significantly improves code aesthetic. Besides that, this coding method is able to encode more information than classical approaches, which open further possibilities of applications.

Furthermore, we will present the pipeline for decoding a graphic code from a photo. It is performed by assessing some images so that coded message is recovered, and it might be supported by using of data redundancy and check digits to validate it, what provides a superior robustness to the whole process.

Figure 1: Combination of a colour pixel-based code (girl) with an icon-based code (musical notes).

1 Introduction

Graphic Code is a Machine-Readable Code (MRC) pattern that has presented significant advances in terms of code aesthetics and the large amount of information that can be encoded in it. This pattern was initially presented by Patrão et al. [2] as a smart marker for Augmented Reality applications. Then, the coding process was deeply described by Cruz et al. [1].

This pattern consists of an appropriate primitive graphic units distribution throughout a certain image. We will assume that these primitive are black or white pixels (although they may be pixels of other colours, or even drawings with some complexity, as presented in Figure 1). Such primitives, the graphic units, are distributed throughout the code into clusters called cells. Some cell patterns are previously associated with characters, forming what is called a dictionary (the primary element of encoding and decoding).

The encoding process returns an image. For practical purposes, this image is printed and must be decoded from a photograph. From this photo, we need to reconstruct the initially generated code to return this code to the decoding process. In this work, we will focus on this reconstruction process. In general, this step consists of properly identifying each primitive graphic unit and its cell patterns.

2 Coding

The basis of our coding and decoding process is a mapping between symbols and patterns that we call dictionary. It defines an alphabet which can be used to encode and decode the message. This alphabet can be binary (0 and 1), numeric (from 0 to 9), alphanumeric (characters from A to Z, numbers from 0 to 9), etc. Our work is a dithering-based approach [4]. This technique converts each pixel of a grayscale image into $k \times k$ black or white pixels. This process reduces the colour space of the image (from 256 levels of grey to black and white) in order to increase image resolution (multiplies each dimension by k) and to preserve perception of grayscale (through colour integration of a cell made by human eyes). According to the dithering technique, each grayscale interval is associated with a specific pattern. This approach can be easily extended to coloured images just by shifting colours to a specific colour-base distribution (when pixels are seen together, they produce original colour perception). We define specific patterns and associate them with symbols.

Coding process receives as input (i) a dictionary, (ii) a base image (which determines the size and overall appearance of the code) and (iii) a message that will be encoded. It creates an image in which each pixel refers to a graphic unit. We will assume that the graphic units (black or white pixels) are grouped into 3×3 cells, this pipeline is illustrated in Figure 3. In addition, we also assume that we will add a checker digit to the end of the message consisting of 3 characters that will be used to validate the decoding process.

After generating the encoded image, we add a specific border, as shown in Figure 2. This border is essential to the reconstruction process, which will be discussed below.

Decoding process, in turn, consists of scanning the code searching in order to reach cell patterns that form the employed dictionary. When such a pattern is found, its corresponding character is added to the message being retrieved.

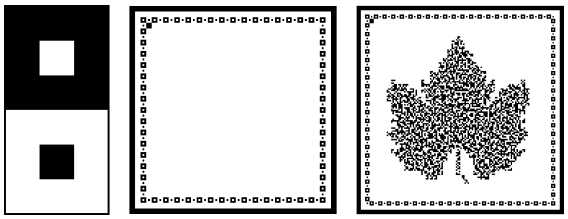


Figure 2: Frame used to support graphic code reconstruction and decoding from a photo.

3 Graphic Code Reconstruction

Reconstruction process is described in Figure 4. It begins with the printed code photo acquisition. Then, we detect the code in this image, and in that region, we identify some points characteristics of the frame. Since our frame is rectangular, we detect in this region the inner and outer quadrilaterals of the external black rectangle of the frame. These quadrilaterals are illustrated in Figure 4 by a red and green polygon. It is noteworthy that, within the region detected as code there are several quadrilaterals,

[1] Patent Application: INPI 20171000063377

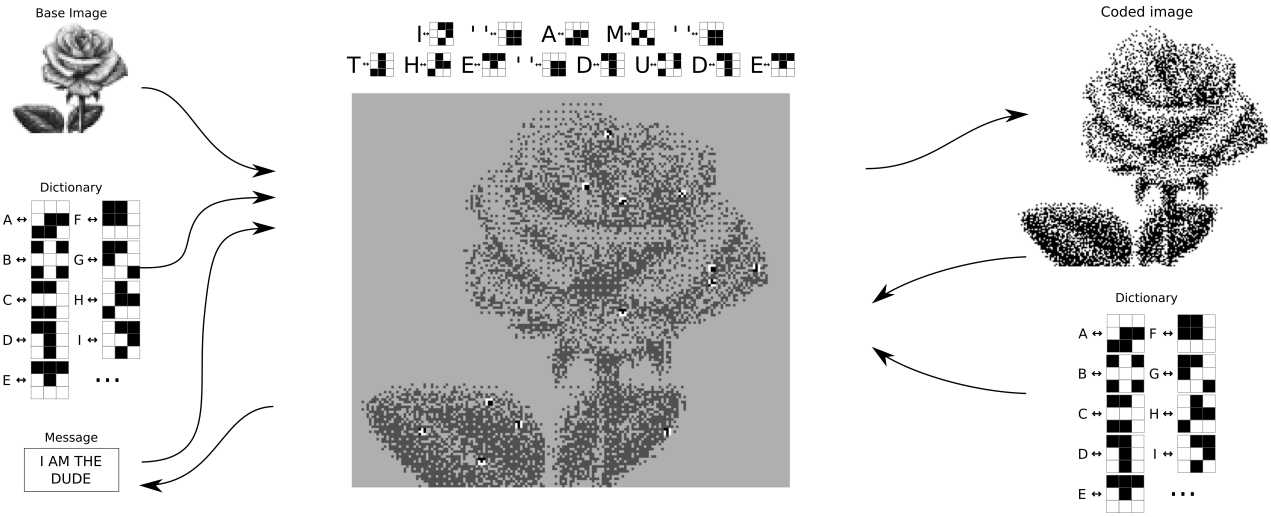


Figure 3: The encoding pipeline starts with a base image, a dictionary and a message. Next, it defines the candidate pixels according to the quanta used in the dictionary, then it places the respective pattern of each message symbol and the encoding finishes by filling the remaining cells while decoding is the opposite process.

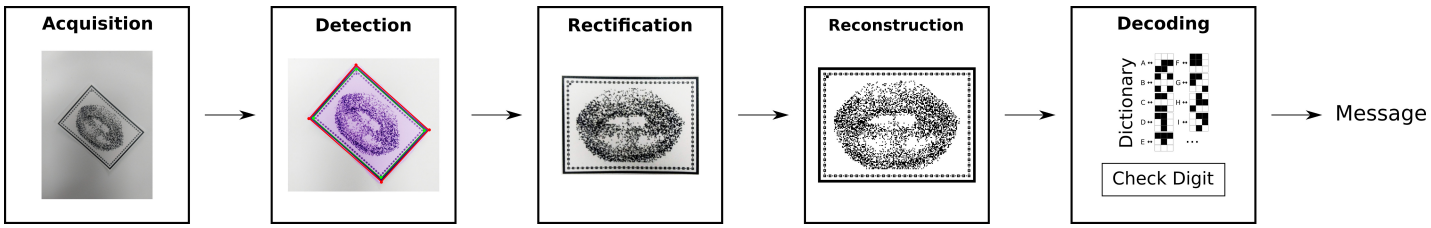


Figure 4: Decoding pipeline process from photos. Starting with photo acquisition, code detection, image rectification, code reconstruction and message decoding.

therefore, this element choice is made by taking the two concentric contours that have the largest area and their respective points.

After retrieving these points, we can rectify the image. The rectification purpose is to obtain an image in which the code appears without distortion and with a known scale. It is achieved by defining an homography matrix that establishes the relationship that transforms the two quadrilaterals obtained in the detection process in rectangles with the expected dimensions. We also use the 3×3 black square at top-left corner of the frame to define the code orientation.

With the rectified image, we proceed with the reconstruction process, that is, identification of each graphic unit. Recognition process of the graphic unit is done as follows: (i) identification of the position in which the samples will be collected and (ii) graphic unit colour assessment.

The rectified code image is 9 times larger in each dimension than the original one. In this way, we have an area of 9×9 pixels referring to the same graphic unit. In other words, the reconstruction process consists of choosing some of these pixels and defining graphic unit colour.

It is important to highlight that although the graphic units are regularly arranged in the initial code (pixels of an image), the printing process, paper curl, photograph artefacts, or inaccuracies in the rectification process create deformations in this rectified image. For this reason, they cause irregular sampling spacing between graphic units and produce such an undesirable target result.

As a result, we perform a correction in the position of these samples from an analysis of the image deformation in the frame edge pattern region. Since the edge pattern is known, we can identify each pixel deformation. Then, it is interpolated to the interior of the model, hence we achieve a better sampling of the graphic units. Such approach has provided impressive results when the printed code lightly curled.

When the position of the samples is decided, we collect 9 of them and apply a voting process to select graphic unit colour. Since the graphic units are black or white, and these samples from the photo are usually grayscale, we need to apply a threshold to decide their colour. This threshold is determined with an analysis of the edge patterns of alternating squares (where there is a properly balanced black and white pixel distribution) that belong to the frame.

4 Concluding remarks

Graphic Code has played such an important role regarding tag patterns and smart markers used in Augmented Reality applications. In this work, we aimed to present how we can decode it from a printed version photograph.

Coding process has a linear complexity ratio. It is particularly accomplished between 20ms and 30ms for models with 60×40 cells of 3×3 graphic units. Similarly, reconstruction process complexity is linear, but it may not have a good reconstruction efficiency at first due to previously identified limitations. However, it has been empirically noticed that when the printed code is lightly curled and the photograph is properly in focus, it does not take more than 10 trials, hence the process is concluded in less than 3 seconds.

Afterwards, the resulting code is translated by the decoding process. Then we verify whether the check digit of the first characters matches the last three ones. When they do not match, it means the reconstruction process failed at some point, and it starts processing the next frame.

Future work points out towards insert redundancy into the graphical units [3] and use it to become reconstruction and decoding processes even more robust. Another future work is to present the reconstruction to other kinds of graphic units (like more general drawings).

References

[1] Leandro Cruz, Bruno Patrão, and Nuno Gonçalves. Halftone Pattern: A New Steganographic Approach. *Eurographics - Short Papers*, 2018.

[2] Bruno Patrão, Leandro Cruz, and Nuno Gonçalves. An application of a halftone pattern coding in augmented reality. *SIGGRAPH Asia Posters*, 2017.

[3] Inving Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 1960.

[4] Robert Ulichney. The void-and-cluster method for dither array generation. *Human Vision, Visual Processing, and Digital Display*, 1993.

Paper:#9

Improving Facial Depth Data by Exemplar-based Comparisons

By: Dihl, L. Cruz, L. Gonçalves, N.

Improving Facial Depth Data by Exemplar-based Comparisons

Leandro Dihl[1]
leandro.dihl@isr.uc.pt
Leandro Cruz[1]
lmvcruz@isr.uc.pt
Nuno Gonçalves[12]
nunogon@deec.uc.pt

[1] Institute of Systems and Robotics
University of Coimbra
Coimbra, Portugal

[2] Portuguese Mint and
Official Printing Office
Lisboa, Portugal

Abstract

3D face models are widely used for several purposes, such as biometric systems, face verification, facial expression recognition, 3D visualization, and so on. They can be captured by using different kinds of devices, like plenoptic cameras, structured light cameras, time of flight, etc. Nevertheless, the model generated by all of these consumer devices are quite noisy. In this paper, we present a filtering method for meshes of faces preserving their intrinsic features. It is based on an exemplar-based neighborhood matching where all models are in a frontal position avoiding rotation and perspective. Moreover, the model is invariant to depth translation and scale. Findings reveal that this method is robust and promising.

1 Introduction

In the last years, it has been increased the number of applications which employ faces captured from consumer 2.5D cameras. A relevant issue is in terms of personal recognition and security. Despite of the 2.5D face acquisition technologies have improved, the resulting outputs are still noisy. Deformed and noisy meshes, holes, and errors are common. These problems impact on 3D visualization processes, face verification models, person recognition, pose detection and facial expression recovery. The outputs of these devices need to be even more improved in order to guarantee reliable and satisfactory results by preserving their geometrical structure.

In this sense, we are developing a specific filter for face meshes, that is, a content-aware method that corrects each point of a given mesh by comparing its neighborhood with the respective neighborhoods in a set of previously defined exemplars. We take advantage of the former knowledge of the models (in this case, the faces) to improve this comparison. In this way, one begins by detecting the facial features points (FFP) and splits the face area according to these points. Eventually, for each FFP region we create a predictor that will be used to filter the points inside this region by matching neighborhoods.

2 Related Work

Many methods have been proposed for denoising and smoothing meshes to improve the output from 3D scanners or 3D cameras. They are usually local and iterative on the mesh data structure. Their main weakness is that they are designed for generic structures not taking into account geometric intrinsic features of the models. Yagou et al. [7] use the mean and a median filter applied to the normal vector of triangles. In this method, the normal vector of a triangle is changed according to the one of its neighbors through the respective vertices position updating. Other methods [1, 2, 8] also perform mesh denoising by normal filtering using a variant of a bilateral filter general formulation. Due to its simplicity and feature-preserving capability, bilateral filter has been used in several image processing applications.

3 Model-Exemplars Neighborhood Comparison

Given a facial RGBD model, the filtering method replaces the depth of each point through a comparison between its neighborhood and the ones of the points into the previously provided exemplars. Figure 2 illustrates the pipeline.

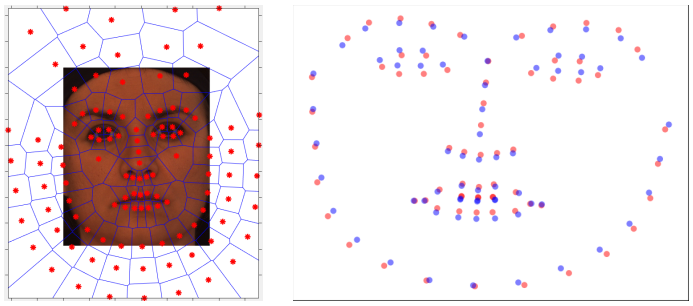
The proposed method is based on a model covering, created from a set of Facial Feature Points (FFP)(Section 3.1). For filtering purpose, it is important to ensure a proper comparability between neighborhood of points in the model and the points in the exemplars. We achieve this goal whereby an FFP alignment followed by a model resampling using the same frequency of the exemplars (Section 3.2).

The proper frequency sampling allows the neighborhood comparison of different models (including models from different types of sensors). Moreover, FFP allows subdividing the facial area in regions that constrain the filtering comparisons into corresponding region. As a consequence of such constraint, we accomplish a coherent correction (facial point matching only uses points of the same part of the face) and computational cost reduction (we only consider points into the specific region instead of all points in exemplars).

In the normalized depth, the filtering procedure is applied to the resampled model (Section 3.2). The resampling guarantees a model resolution invariance, so as to the FFP alignment provides translation, rotation and scaling invariance, and the neighborhood normalization implies depth translation and scaling invariance.

3.1 Face Covering and Alignment

As aforementioned, facial covering is based on FFP. It is composed by the following steps: (i) detection, (ii) insertion, and (iii) removal of FFP; (iv) creation of Voronoi diagram with these points, and finally (v) the definition of points as a whole that belong to each region.



(a) (b)

Figure 1: On the left, (a) we illustrate FFP estimation (red points) and facial region division (blue polygons); on the right, (b) we illustrate the alignment of two sets of FFP (red and blue points).

The subdivision process shall meet three conditions:

1. All faces must have FFP at correspondent places;
2. The union of regions must cover the whole face;
3. The area of each region must be inversely proportional to the average of expected noise.

Our FFP calculation is based on the method proposed by Kazemi and Sullivan [4], which was adapted to achieve these three conditions. After points detection [4], we insert new ones in large areas with lack of feature points, such as the center of the cheeks and forehead. These points are generated by using geometric relations from former resulting points. At this step, needless points such as the central lip contours are removed to prevent exceeding search regions generation.

The second step of face coverage is Voronoi diagram creation, which corresponds to a special kind of decomposition of a given space, for example, a metric space, determined by the distance between the FFP. Figure 2(a) illustrates the Voronoi diagram obtained from the FFP considered in the previous steps.

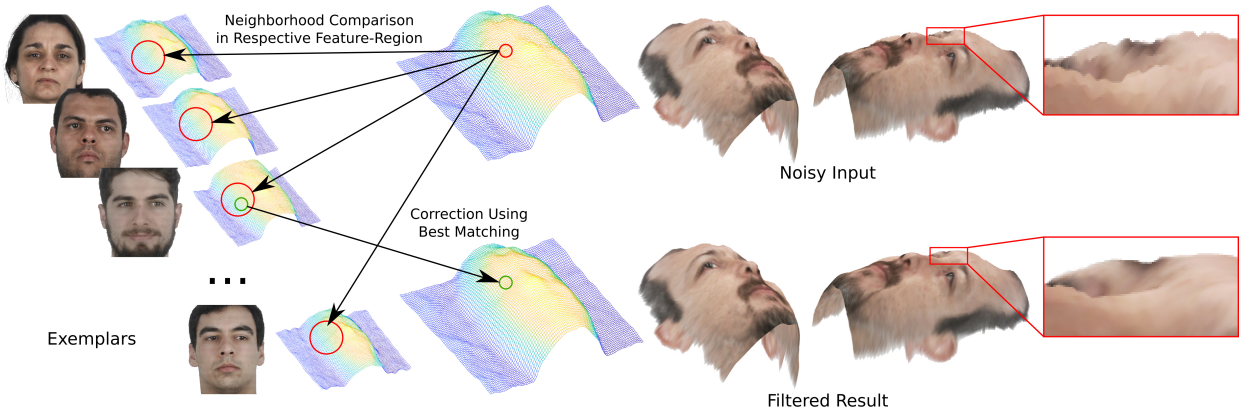


Figure 2: The pipeline of the proposed method.

Finally, the last step corresponds to the definition of the R search regions that are defined by the polygons generated in the Voronoi diagram. Each region is expanded in order to avoid distortions in the filtering step.

To properly guarantee a sampling frequency of all exemplars and models, we perform an alignment [5] of their FFP. First, we choose one exemplar (whichever) to be the basis, then we align all others according to this one. After the alignment, we scale each face with respect to the basis one by using the alignment affine transformation [5]. Finally, we can resample the other face by the same frequency sampling of the base one. Figure 2(b) shows alignment of the faces.

3.2 Normalized Depth-Neighborhood Comparison

For each point p in the mesh, we create its correspondent neighborhood as a $k \times k$ (k is odd) matrix in which the central value is the depth of p and the other values are the depth of the respective neighbor. The comparison between two neighborhoods is performed with Euclidean distance. However, even when we are comparing two geometrically similar neighborhoods, it is possible there exists a significant difference. To achieve a proper measure of geometric similarity, we employ a pattern in all neighborhoods as follows:

$$\Phi(x,y) = \frac{\psi(x,y) - \mu}{\sigma} \tag{1}$$

where μ is its mean and σ is its variance.

The standard step purpose is handling the models obtained with different setups (such) as distance and scale making the approach more robust. In this way, μ creates a translational invariance and σ creates an invariance to scale (both in the depth).

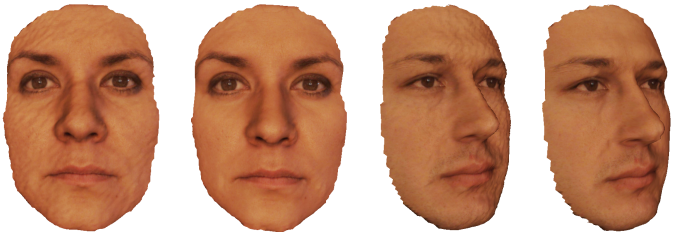
We denote M_p by the point in the model in its respective position p, and E^e_q by the point q in the exemplary e. Neighborhood of each point M_p is compared to the ones in the exemplars, thus depth is replaced by the best matching depth. Search for the most similar neighborhood on exemplars is performed by a Nearest Neighbor method, implemented with the usage of a Kd-Tree together with PCA for dimensionality reduction [3]. The neighborhood comparison metric of a point M_p, and a point E^e_q is given by:

$$dist\left(M_p, E^e_q\right) = \sum_{i=1}^{k}\sum_{j=1}^{k}\left(M_p(i,j) - \left(E^e_q(i,j)\right)\right)^2 \tag{2}$$

4 Results

The initials results were generated using a set of exemplars based on the Bosphorus Database[6]. This database is to research on 3D and 2D human face processing tasks including expression recognition, facial action unit detection, facial action unit intensity estimation, face recognition under adverse conditions, etc.

In order to achieve the results, we performed the experiments applying noise to the set of the different models. Figure 3 shows the visual achieved results. (a) and (c) are the noised meshes and (b) and (d) are the filtered images.



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

Figure 3: Some visual results obtained with the proposed method. (a) and (c) Noised meshes. (b) and (d) Filtered images

5 Conclusions and Future Work

We presented a content-aware model for face meshes filtering based on exemplars. In addition, early results robustness seemed to be promising. Another approach for the future lies on establishing a validation metric for the results. It might be based on the measure of the variance of each point. Indeed, the proposed method reduces the overall range. It is also possible to use the symmetry of faces in order to create such quality measure. Furthermore, we can define a metric by a comparison of the filtered model with a facial template deformed by the FFPs.

Finally, our filtering approach is based upon a division of the model in regions in which all points have an intrinsic geometric similarity. We described how to define these regions for the specific case of faces by means of FFPs. For other types of models, it is necessary to use a feature detector that meets the three conditions presented in Section 3.1. A future work direction is to define general descriptors that might be used for general purpose filtering.

References

[1] Mario Botsch, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Lévy, Stephan Bischoff, and Christian Rössl. Geometric modeling based on polygonal meshes. In *ACM SIGGRAPH Courses*, 2007.

[2] Hojin Cho, Hyunjoon Lee, Henry Kang, and Seungyong Lee. Bilateral texture filtering. *ACM Trans. Graph.*, 33(4):128:1–128:8, 2014.

[3] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *SIGGRAPH*, 2001.

[4] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE CVPR*, 2014.

[5] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, Dec 2010.

[6] Arman Savran and Blent Sankur. Non-rigid registration based model-free 3d facial expression recognition. *Comput. Vis. Image Underst.*, 162(C):146–165, September 2017. ISSN 1077-3142.

[7] H. Yagou, Y. Ohtake, and A. Belyaev. Mesh smoothing via mean and median filtering applied to face normals. In *Geometric Modeling and Processing. Theory and Applications*, 2002.

[8] Y. Zheng, H. Fu, O. K. Au, and C. Tai. Bilateral normal filtering for mesh denoising. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1521–1530, Oct 2011.

Paper:#10

Unveiling Markers of Stress Via Smartphone Usage

By: Sharma, R. Ribeiro, B. Pinto, A. Cardoso, F. Armando, N. Raposo, D. Silva, J. Oliveira, H. Macedo, L. Boavida, F. Fernandes, M. Rodrigues, A.

Unveiling Markers of Stress Via Smartphone Usage

Rahul Sharma[1], Bernardete Ribeiro[1], Alexandre Miguel Pinto[1], F. Amílcar Cardoso[1], Ngombo Armando[1,3], Duarte Raposo[1], Marcelo Fernandes[1], André Rodrigues[1,2], Jorge Sá Silva[1], Hugo Gonçalo Oliveira[1], Luis Macedo[1], Fernando Boavida[1]

[1] CISUC, University of Coimbra, Portugal.
[2] ISCAC, IPN Coimbra, Portugal.
[3] ESPU, Universidade Kimpa Vita, Angola.

[rahul|bribeiro|ampinto|amilcar|narmando|draposo|arodz|sasilva|hroliv|macedo|boavida]@dei.uc.pt,jmfernandes@student.dei.uc.pt

Abstract

Numerous Android apps leverage from the information provided by embedded sensors of the smartphones. The prime objective of this work is to conduct a state-of-the-art short survey of stress-related research and determine which inbuilt sensors and features of smartphone applications can help in determining stress among students. The study focused on three factors, physical activities, sociability, and ambiance, and it shows how smartphones can take advantage of these aspects to determine stress.

1 Introduction

Stress is, any uncomfortable "emotional experience accompanied by predictable biochemical, physiological and behavioral changes" [3]. Factors of Stress are omnipresent that can influence anyone irrespective of their gender, age, living conditions, etc., and can trigger physical issues (like muscle pain, high blood pressure, and a weakened immune system) and psychological problems (such as depression and anxiety). Besides physiological symptoms, over-consumption of junk food, alcoholism, drugs, and smoking are a few behavioral indicators of stress [9]. Although stress is a psychological phenomenon, it has physiological impacts such as variation in skin conductance, neck pain, blood pressure, heart rate, catecholamine, and cortisol secretions [18]. All of these phenomena can be used to evaluate stress level either clinically or technologically, and with high credibility, but are not very practical due to cost, test durations, user biasedness and truthfulness in case of surveys. The idea of this work, i.e., detecting stress through smartphones, was conceived while investigation inactivity and activity among the students via smartphone usage in the project SOCIALITE[1] [2], Section 2 puts forward the motivation of behind work. Further, state of the art related to three markers of stress (i.e., Physical Activities, Sociability, and Ambiance) are proposed, along with the probable features and evaluators that can be captured via smartphones. In Section 3 concluding remarks are made.

2 Survey over Markers of Stress

This section describes the motivation of our contribution. The probable three factors that influence the stress levels in humans are also inspected in this section, along with the three types of data collection via a smartphone app as shown in Figure 2.

•**Elemental Work with Smartphone** dentification of stress-related

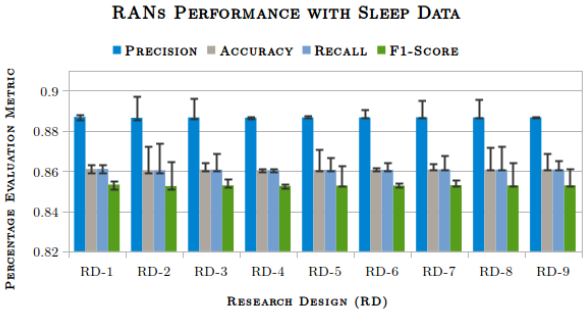


Figure 1: Evaluation of Modeling of SOCIALITE data with RANs

attributes via smartphone has its motivation from an experiment of project SOCIALITE. The project SOCIALITE is an attempt to address some aspects related to IoTs, by considering the existing IoT framework (such as FIWARE) in conjunction with people-centric technologies (like smartphones, sensor-boxes, and other IoT source), and proposing a solution consisting WSNs, mobility, and ubiquity, along with cognitive services and context-awareness, for supporting People-2-People interaction. Nowadays, smartphones are omnipresent and have been an invaluable source of data. To benefit from this smartphone data, in project SOCIALITE, smartphone app ISABELA[2] [2] was used to capture data pertaining to the attributes such as *Activity, Day, Luminescence, Sound, Alarm, and Phone-lock-state*. The data was then utilized to model the "Active-State", and "Inactive-State" of the recipients using a computational modeling approach Regulated Activation Networks [15]. The generated model was evaluated with Precision= 89.02% (ca.), Recall= 86.13% (ca.), F1-score= 85.53% (ca.), and Accuracy= 86.13% (ca.) (see Figure 1 for the research design with varying train data for 90% -to- 10%, and vice versa for test data). Further, statistical analysis of observations of data (specific to three students individually) with the generated model depicted that one student was mostly "Inactive" all the time, which does not commensurate with the assumed normal behavior of the student [15]. Consequently, it indicates the reasons (such as personal problems, illness, or stress) that leads to such behavior of that student.

Figure 2: Data acquisition of markers of stress via a smartphone app

• **Physical Activities and their impact on stress**. Activities involving the physical movement of the human body has an immense effect in relieving stress [13]. There are pieces of evidence that exercises, frequent walks, swimming helps in increasing the production of feel-good endomorphins that help in treating mild forms of depression and anxiety [6]. Physiological movements are also found to alleviate hypertension, blood pressure, and obesity-related problems, which are among the critical symptoms of stress [10]. In the Internet of Things (IoT), Human Activity Recognition (HAR) is an essential domain of research with a notable contribution focusing upon identification of human activities such as walking, running, jumping, and sitting [1]. Since physical movement is an essential factor that correlates with the stress and attributes related to physical movements can be utilized in determining stress levels, see first row of Table 1 for probable features that can be observed via Smartphone.

• **Social communications and its relation with stress**. Being social is an essential part of a healthy lifestyle, however, it is not necessary to be very eloquent in order to be social, as communication is a crucial part of sociability, and highly correlated to loneliness and social isolation [14] can motivate people to adopt ill habits such as smoking, drinking, and drug usage [12]. Smartphones have been used to study the sociability of students by monitoring their physical activity and ambient noise [8]. Sociability is necessary, but there are shreds of evidence that it is related to the decline in the academic performance of students [7], this deterioration has been linked to stress among students [17]. With the aid of data from app usage of smartphones it's possible to study sociability of a person and determine how it is linked to stress, see second row of Table 1 for possible Sociability related attributes.

• **Effects of environment over stress**. As aforementioned, stress

[1] "Social-Oriented Internet of Things Architecture, Solutions and Environment"- https://www.cisuc.uc.pt/projects/show/215

[2] IoT Student Advisor and BEst Lifestyle Analyzer (ISABELA)

Table 1: Proposed Attributes for all 3 markers of stress

Marker	Atributes	Labels
Physical Activity	Walking, Running, Sitting, Laying, Jumping, Standing	Image Stress Meter, Stroop Color Test , Stress Questionnaire
Sociability	Facebook, Whatsapp, Twitter, Voice call, Text messages	Image Stress Meter, Stroop Color Test , Stress Questionnaire
Ambience	Noise, Temperature, Humidity, Pollutants	Image Stress Meter, Stroop Color Test , Stress Questionnaire

is a phenomenon that emanates from an individual's evaluation and response to its environment. The Ambiance is an important determinant that can influence the stress accumulation in an individual. Crowd, noise, climate, and pollution are a few such factors that have an impact on the psychological conditions of a person [5]. Smartphones embedded sensors can be used to determine all four, prior, mentioned stressors. The microphone can be used to determine the number of different speakers in the crowd [19], and the noise as in the environment [20]. Ambient pollutant and climate-related data can be obtained from global weather data APIs, see the third row of Table 1 that shows the Ambience related feature that can be collected.

The Table 1 lists the attributes and labels that can be collected via smartphone w.r.t three Markers. The data representing characteristics, logged in Table 1, can be gathered in an aggregated manner, i.e. average values of the attributes to be collected at a fixed interval of time. In the case of Physical Activity attributes, the average time spent in performing the activity should be recorded for the period. Whereas the Sociability attribute can log minutes spent on texting, talking, and browsing during the interval, and count of people connected with the user. For Ambiance related characteristics, average, minimum, and maximum values can be collected. To establish the relationship between the collected data for the three stressors mentioned above with stress levels of an individual, one of the following three stress tests are recommended for recording the stress levels of the subject as the label. Perceived Stress Scale (PSS) [4] is a classic stress assessment instrument and can be used as a reference to develop the questionnaire. Photographic Affect Meter [11] is a tool to associate emotion with images and identify stress in humans. Stroop's Effect [16] is also suitable for the stress evaluation purpose, as it has vast literature on evaluating criteria, and easy to implement as an app.

3 Conclusion

Stress is an important phenomenon and primarily studied through psychological, and biological methods. This article produces a shot survey of three important factors (Physical activities, sociability, and ambiance) that have been correlated with stress. Some contributions tend to determine attributes related to, previously mentioned, 3 aspects with the aid of smartphones. These attributes have been beneficial for research such as mobility detection, and environmental monitoring. In the scope of SOCIALITE project, an initial work shows how it is possible to reveal stress levels of a subject by modeling "Activity" and "Inactivity", and statistically analyzing them. Furthermore, as future work, the data obtained from smartphones about these characteristics will help not only in a learning system that determines personalized stress models, but also enables monitoring, and effectively manage stress among people.

Acknowledgement

The work was performed under the scope of the SOCIALITE Project (PTDC/EEI- SCR/2072/2014), co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Inter- nationalization (POCI), European Union's ERDF (European Regional Development Fund), and the Portuguese Foundation for Science and Technology (FCT).

References

[1] Alvina Anjum and Muhammad Usman Ilyas. Activity recognition using smartphone sensors. In *Consumer Communications and Networking Conference (CCNC)*, pages 914–919. IEEE, 2013.

[2] Ngombo Armando, Duarte Raposo, Marcelo Fernandes, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. Wsns in fiware–towards the development of people-centric applications. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 445–456. Springer, 2017.

[3] Andrew Baum. Stress, intrusive imagery, and chronic distress. *Health psychology*, 9(6):653, 1990.

[4] Sheldon Cohen, T Kamarck, R Mermelstein, et al. Perceived stress scale. *Measuring stress: A guide for health and social scientists*, pages 235–283, 1994.

[5] Gary W Evans. *Environmental stress*. CUP Archive, 1984.

[6] Kenneth R Fox. The influence of physical activity on mental well-being. *Public health nutrition*, 2(3a):411–418, 1999.

[7] Fausto Giunchiglia, Mattia Zeni, Elisa Gobbi, Enrico Bignotti, and Ivano Bison. Mobile social media usage and academic performance. *Computers in Human Behavior*, 82:177–185, 2018.

[8] Gabriella M Harari, Samuel D Gosling, Rui Wang, Fanglin Chen, Zhenyu Chen, and Andrew T Campbell. Patterns of behavior change in students over an academic term: A preliminary study of activity and sociability behaviors using smartphone sensing methods. *Computers in Human Behavior*, 67:129–138, 2017.

[9] Alan E Kazdin et al. *Encyclopedia of psychology*, volume 8. American Psychological Association Washington, DC, 2000.

[10] Husein Mohammed, Shibani Ghosh, Fred Vuvor, Seth Mensah-Armah, and Matilda Steiner-Asiedu. Dietary intake and the dynamics of stress, hypertension and obesity in a periurban community in accra. *Ghana medical journal*, 50(1):16–21, 2016.

[11] John P Pollak, Phil Adams, and Geri Gay. Pam: a photographic affect meter for frequent, in situ measurement of affect. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 725–734. ACM, 2011.

[12] Douglas A Raynor and Heidi Levine. Associations between the five-factor model of personality and health behaviors among college students. *Journal of American College Health*, 58(1):73–82, 2009.

[13] Peter Salmon. Effects of physical exercise on anxiety, depression, and sensitivity to stress: a unifying theory. *Clinical psychology review*, 21(1):33–61, 2001.

[14] Aparna Shankar, Anne McMunn, James Banks, and Andrew Steptoe. Loneliness, social isolation, and behavioral and biological health indicators in older adults. *Health Psychology*, 2011.

[15] Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, and F. Amílcar Cardoso. perceiving abstract concepts via evolving computational cognitive modeling. In *International Joint Conference of Neural Networks*. IEEE, 2018.

[16] J Ridley Stroop. Studies of interference in serial verbal reactions. *Journal of experimental psychology*, 18(6):643, 1935.

[17] C Ward Struthers, Raymond P Perry, and Verena H Menec. An examination of the relationship among academic stress, coping, motivation, and performance in college. *Research in higher education*, 41(5):581–592, 2000.

[18] Joachim Taelman, Steven Vandeput, Arthur Spaepen, and Sabine Van Huffel. Influence of mental stress on heart rate and heart rate variability. In *4th European conference of the international federation for medical and biological engineering*. Springer, 2009.

[19] Chenren Xu, Sugang Li, Gang Liu, Yanyong Zhang, Emiliano Miluzzo, Yih-Farn Chen, Jun Li, and Bernhard Firner. Crowd++: unsupervised speaker count with smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 43–52. ACM, 2013.

[20] Jinbo Zuo, Hao Xia, Shuo Liu, and Yanyou Qiao. Mapping urban environmental noise using smartphones. *Sensors*, 16(10):1692, 2016.

Paper:#12

Understanding Deep Neural Networks decisions in Medical Imaging

By: Silva, W. Fernandes, K. Cardoso, M. Cardoso, J.

Understanding Deep Neural Networks decisions in Medical Imaging

Wilson Silva[1,2]
wilson.j.silva@inesctec.pt

Kelwin Fernandes[1]
kafc@inesctec.pt

Maria J. Cardoso[2,3,4]
maria.joao.cardoso@fundacaochampalimaud.pt

Jaime S. Cardoso[1,2]
jaime.cardoso@inesctec.pt

[1] Faculdade de Engenharia,
Universidade do Porto, Porto, Portugal

[2] INESC TEC
Porto, Portugal

[3] Faculdade de Ciências Médicas,
Universidade NOVA de Lisboa, Lisboa, Portugal

[4] Unidade de Mama,
Fundação Champalimaud, Lisboa, Portugal

Abstract

Interpretability is a fundamental property for the acceptance of machine learning models in highly regulated areas. Recently, deep neural networks gained the attention of the scientific community due to their high accuracy in vast classification problems. However, they are still seen as black-box models where it is hard to understand the reasons for the labels that they generate. This paper proposes a deep model with monotonic constraints that generates complementary explanations for its decisions both in terms of style and depth. Furthermore, an objective framework for the evaluation of the explanations is presented. Our method is tested on a post-surgical aesthetic evaluation dataset and demonstrates an improvement in relation to traditional models in terms of quality of the explanations generated.

1 Introduction

In the literature, it is possible to find several different approaches for the generation of explanations that justify the model's decision, and are, at the same time, understandable to a human being.

Kim and Doshi-Velez [2] summarize the different interpretability approaches in three domains: pre-model, in-model and post-model.

Pre-model interpretability consists on the understanding of the data itself through visualization and exploratory data analysis. For instance, one can easily interpret a complex data distribution considering prototypical examples, if the dataset is representative enough.

In-model interpretability is composed of strategies that can be embedded in the machine learning model to increase its interpretability. Models based in rules, cases, sparsity and/or monotonicity are included in this category.

Finally, post-model interpretability is constituted by the strategies that try to understand the machine learning model's behavior after the model has been built. In here, sensitivity analysis, mimic models and investigation on hidden layers of DNN are possible approaches.

2 Complementary Explanations using Deep Neural Networks

Inspired by the concept of different types of learners [3], we present here a deep model capable of providing different styles of explanations.

A DNN is able to integrate more than one interpretability strategy, namely, case-based (in-model), monotonicity (in-model) and sensitivity analysis (post-model). Thus, we make use of that capability and create the DNN model described in Fig. 1. Our model has two streams, one that has as input the already monotonic features, and another that takes as input non-monotonic features and has the goal of transforming them into a monotonic latent space. All monotonic blocks are composed of dense layers with positive constraints regarding their weights. To generate the explanations, two strategies are followed: explanation by local contributions (sensitivity analysis) and explanation by similar examples (case-based).

2.1 Explanation by local contribution

Explanations by local contribution consist on the selection of the features with higher impact on the decision. The measurement of the contribution, C_{ft}, of a feature ft is found through an adversarial example, i.e., we search for the value X_{opt} that approximates X to the opposite class, $\bar{y}$. Eq. 1 describes the loss function to be minimized, where $\bar{y} = 1 - y$ (we are restricting the problem to a binary scenario), $y \in \{0,1\}$, and $f(X)$ is the estimated probability. The minimization is obtained using backpropagation with respect to the feature, ft.

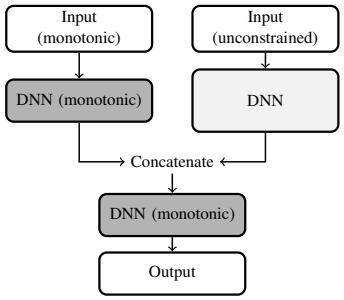


Figure 1: Proposed DNN architecture.

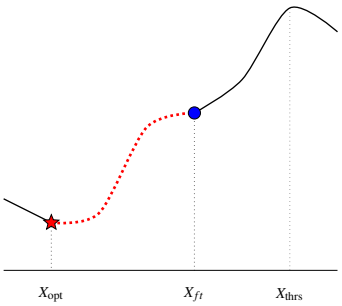


Figure 2: Feature impact analysis.

$$(\bar{y} - f(X))^2 \tag{1}$$

To avoid repetitive explanations, the contribution of each feature, C_{ft}, on the target variable is weighted by the range of the feature domain traversed from the initial value, X_{ft}, to the local minimum, X_{opt} (Fig. 2). The process is described by Eq. 2, where X' is the input vector after assigning X_{opt} to ft. In its turn, the inductive rule constructed for ft covers the space between X_{ft} and X_{thr}, the value where the probability of the predicted class is maximum.

$$C_{ft} = |f(X) - f(X')| \cdot \frac{X_{ft} - X_{opt}}{X_{\max} - X_{\min}} \tag{2}$$

2.2 Explanation by similar examples

Explanations by similar examples are generated finding the nearest neighbors in a learned semantic space. Again, we can use sensitivity analysis to evaluate which features have the major impact on the semantic space distance between two observations. Therefore, two types of explanations can be inferred: why an observation is similar to another (nearest neighbor of the same class), and why an observation is different of another one (nearest neighbor of the opponent class).

3 The three Cs of interpretability

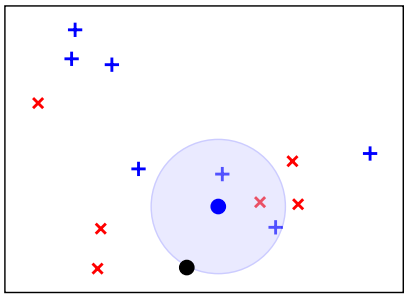
Although there is much work done regarding the creation of interpretable models, there is no objective way of comparing them. In this work, we present some proxy functions that can be used to summarize the quality of an explanation. We named the proxy functions as the "three Cs of interpretability": completeness, correctness, and compactness. Completeness measures the generality of an explanation, correctness its accuracy and compactness its size. Fig. 3 and Fig. 4 help with the understanding of these functions. Completeness is defined as the ratio of observations that are included by the decision rule precondition, or within the same distance of the neighbor explanation, by the total number of observations. Correctness is defined by the label-agreement between the blue rows and between the points inside the n-sphere. Finally, compactness is determined by the number of conditions in the decision rule and the feature dimensionality of a neighbor-based explanation.

$$\texttt{If } A \geq 1 \wedge B \leq 5 \wedge B \geq 2 \texttt{ then } y = 1$$

A	B	Y
0	4	1
2	3	1
4	2	0
3	6	1
3	3	1

Completeness: 3/5
Correctness: 2/3
Comptactness: 3

Figure 3: Illustration of explanation quality for decision rules.



Completeness: 4/14
Correctness: 3/4
Comptactness: 2

Figure 4: Illustration of explanation quality for KNN (where the black dot is the new observation and the blue dot is the nearest-neighbor).

4 Experimental Assessment

To assess the performance of the proposed methodology, we considered post-surgical aesthetic evaluation. Compactness is defined here as the size in bytes of the explanation after compression using the standard Deflate algorithm. Thus, lower values are better than higher values. Explanations are generated to account for 95% of the feature impact and embedding distance.

The dataset is composed of 143 images, the ones that had a panel consensus about the aesthetic classification [1]. To predict the classification we considered 23 high-level features describing breast asymmetry in terms of shape, global and local (scars) color differences. Moreover, the classification problem was reduced to a binary classification problem with the following classes: {Excellent} *vs.* {Good, Fair, Poor}, {Excellent, Good} *vs.* {Fair, Poor} and {Excellent, Good, Fair} *vs.* {Poor}. The results presented in Table 1 show a better correctness performance of the DNN in relation to its competitors (decision tree and 1-nearest neighbor) for rule explanations. For case-based explanations, the 1-NN led to better correctness results but at the expense of completeness. Figure 5 shows some explanations produced by the DNN.

5 Conclusions

In this work, we proposed a DNN model and an evaluation framework. The DNN model is able to generate complementary explanations both in terms of type and granularity. The evaluation framework proposed consists on three proxy functions that summarize relevant aspects of interpretability, as the completeness, correctness, and compactness of the explanations.

Table 1: Quality of the predictions in terms of area under the ROC. Quality of the explanations in terms of completeness (Compl), correctness (Corr), and compactness (Compt).

Binarization	Model	ROC	Explanations			
			Type	Compl	Corr	Compt
Excellent *vs.* Good, Fair, Poor	DT	71.96	Rule	3.82	75.52	31.97
	1-NN	67.37	Similar	3.25	89.27	95.94
			Opponent	80.84	72.96	96.00
	DNN	**80.61**	Similar	95.20	85.69	124.94
			Opponent	46.87	92.04	149.68
			Rule	3.69	99.91	62.59
Excellent, Good *vs.* Fair, Poor	DT	85.18	Rule	3.16	51.75	30.00
	1-NN	52.81	Similar	2.98	85.69	95.94
			Opponent	91.26	54.76	95.97
	DNN	**86.78**	Similar	17.34	72.52	80.36
			Opponent	31.28	81.16	138.00
			Rule	2.33	98.89	48.59
Excellent, Good Fair *vs.* Poor	DT	**94.20**	Rule	6.71	76.92	17.08
	1-NN	54.42	Similar	3.01	94.45	95.94
			Opponent	85.33	84.42	96.00
	DNN	91.03	Similar	1.46	87.25	79.79
			Opponent	67.86	92.82	157.81
			Rule	5.48	99.88	58.44

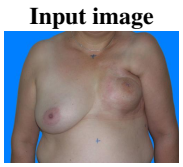


Figure 5: Visualization of the explanations. $\overline{pBOD}$ and $\overline{pBCE}$ represent the negation of the original features, $pBOD$ and $pBCE$, and are presented to make the explanation more intuitive.

The results obtained show an improvement in relation to standard methods regarding the equilibrium between correctness and completeness of the explanations. Moreover, the generated explanations made sense to an expert.

Future work will focus on extending this model to ordinal and multi-class classification. Furthermore, we intend to create an end-to-end model that is able to generate explanations directly from images and that includes other kinds of explanations, as the localization of high impact zones.

6 Acknowledgements

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia as part of project UID/EEA/50014/2013 and within PhD grant number SFRH/BD/139468/2018.

References

[1] Jaime S Cardoso and Maria J Cardoso. Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment. Artificial Intelligence in Medicine, 40(2):115–126, 2007.

[2] Been Kim and Finale Doshi-Velez. Interpretable machine learning: The fuss, the concrete and the questions. ICML Tutorial on interpretable machine learning, 2017.

[3] Harold Pashler, Mark McDaniel, Doug Rohrer, and Robert Bjork. Learning styles: Concepts and evidence. Psychological science in the public interest, 9(3):105–119, 2008.

Paper:#17

Forecasting Household Energy Consumptions using Capsule Networks

By: Leitão, J. Gil, P. Ribeiro, B. Cardoso, A.

Forecasting Household Energy Consumptions using Capsule Networks

Joaquim Leitão[1]
jpleitao@dei.uc.pt
Paulo Gil[2]
psg@fct.unl.pt
Bernardete Ribeiro[1]
bribeiro@dei.uc.pt
Alberto Cardoso[1]
alberto@dei.uc.pt

[1] Center for Informatics and Systems of the University of Coimbra (CISUC), University of Coimbra, Coimbra, Portugal
[2] Centre of Technology and Systems (UNINOVA-CTS) Universidade NOVA de Lisboa, Monte de Caparica, Portugal

Abstract

In this work we apply the novel capsule networks to forecast energy consumptions in a domestic environment. We take consumption data from a real-world dataset to train and test our capsule model.

Despite their improvements over other neural network architectures, namely convolutional networks, predictions computed with our model remain far from the expected values.

These results call for further investigation on the design of the network and definition of its parameters, as well as in data pre-processing, in order to improve forecasting accuracy.

1 Introduction

Buildings are one of the main sources of energy consumptions, responsible for 30 to 45% of the world's global energy consumption [2].

Given the importance of energy to human communities [7] and the growth of its consumption over time, a sustainable and more efficient management of these resources is required. Extensive research has been conducted in this topic, in both small and large-scale scenarios [8].

In the current work we apply a capsule network to forecast domestic energy consumptions. Indeed, the ability to accurately forecast energy consumptions is of great interest, as it allows improved and more efficient management of the electrical grid and its resources.

The remainder of this document is organised as follows: Section 2 reviews time-series forecasting techniques. Section 4 presents our capsule network structure, while Section 3 introduces the historic consumption data used in this work. Finally, Section 5 analyses our model's predictions and Section 6 concludes the document.

2 Forecasting Domestic Energy Consumptions

In classical time-series forecasting techniques, parametrized models like ARMAX or ARIMA are fitted to a sequence of values. While building these models is a relatively simple task, they present strong limitations as non-linear and non-stationary processes cannot be properly modelled.

As a result, various machine learning techniques have been applied to model and predict sequential data [9]. In [1], recurrent neural networks (RNN) and convolutional neural networks (CNN) were highlighted from other techniques in the literature for their accurate time-series forecasts.

Long short term memory (LSTM) networks are the most popular implementation of RNN for time-series forecasting. Among CNNs, we highlight the WaveNet architecture [11] and their dilated causal convolutions, providing two main advantages: (a) assuring the causality between predictions - predictions made at time step t cannot depend on values in future time steps $t+i$, with $i \in \mathbb{N}$; and (b) increasing the receptive field without a substantial grow in the model's depth.

Currently the state of the art in image classification [6], CNNs present one well-known drawback: they fail to encode spatial relationships between the data. Even though research is still being conducted to improve CNNs, other authors have studied innovative alternatives. Initially introduced in 2011 [3], and later formalised in 2017 [10], the CapsNet architecture was proposed to overcome limitations of CNNs. Local *capsules* are the main idea in this neural network, learning implicit features over a limited domain of viewing conditions and deformations. Motivated by the novelty and improvements over CNNs, capsule networks were applied to forecast domestic energy consumptions.

3 Consumption Data

In the scope of this work, domestic energy consumptions were obtained from Pecan Street Inc. Dataport Dataset [4]. 4 years worth of energy readings were collected from one house, from 2014 to 2017 with hourly intervals. The first two years were used for training, the third (2016) for validation, and the last year (2017) for testing. Forecasts are computed based on a window of 24 previous consumptions, determined via an auto-correlation analysis.

Logarithm normalisation was carried out: the logarithm of time-series values was taken, subtracting its mean and dividing them by their standard deviation. Resulting (transformed) time-series values follow a standard normal distribution, $\mathcal{N}(0,1)$.

4 Network Architecture

Figure 1 presents our capsule network architecture. This architecture was mainly inspired on proposals in [5] and [10], where these networks were applied to speed traffic prediction and handwritten digit classification.

Similarly to cited works, inputs are fed to a dilated causal convolution layer. Automatic feature extraction is performed, by learning relevant relations between past elements of the series. A total of 32 filters with a width of 3 and linear activation function were considered, along with $L2$ regularization to prevent overfitting.

Features extracted in this layer are provided to the PrimaryCaps layer, designed similarly to the proposal in [10]: 32 channels of convolutional $8D$ capsules with shared weights, stride of 2 and zero padding.

When applying capsule networks for digit classification, [10] consider one final layer with ten $16D$ capsules, one per digit. Dynamic routing was performed between PrimaryCaps and this layer. Each capsule learns variations in the representations of their assigned digit. The length of each capsule vector encodes the probability of that digit being represented in the input.

The goal of our work is to forecast a future time-series value, based on a window of past inputs. We considered a capsule layer to learn high-level features. Actual forecasts were computed as the output of a fully connected layer, taking the capsules as inputs. We varied the number of capsules in our model. Since more capsules results in longer training times and promotes overfitting, our final *SeriesCaps* layer was composed of five $16D$ capsules.

5 Results

Implementation was conducted using Keras API, running on top of Tensorflow. The model was trained using the Adam optimizer to minimise the mean absolute error (MAE) loss function. The CapsNet model was compared against a CNN model composed of three convolutional layers whose output is connected to a dense layer (linear activation function). Max-pooling between convolutions was also considered.

Figure 4 presents forecasts for the test dataset. Effective energy consumptions, after reverting normalisation, are presented.

Forecasts were evaluated according to the root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). Both RMSE and MAE express average model prediction errors in the same units as the variable being predicted, have no upper bound, but MAE is less sensitive to extreme values.

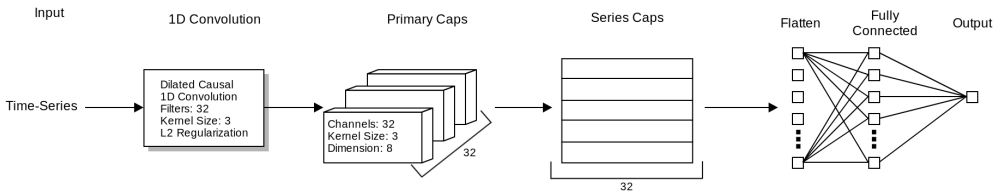


Figure 1: Architecture for our capsule network.

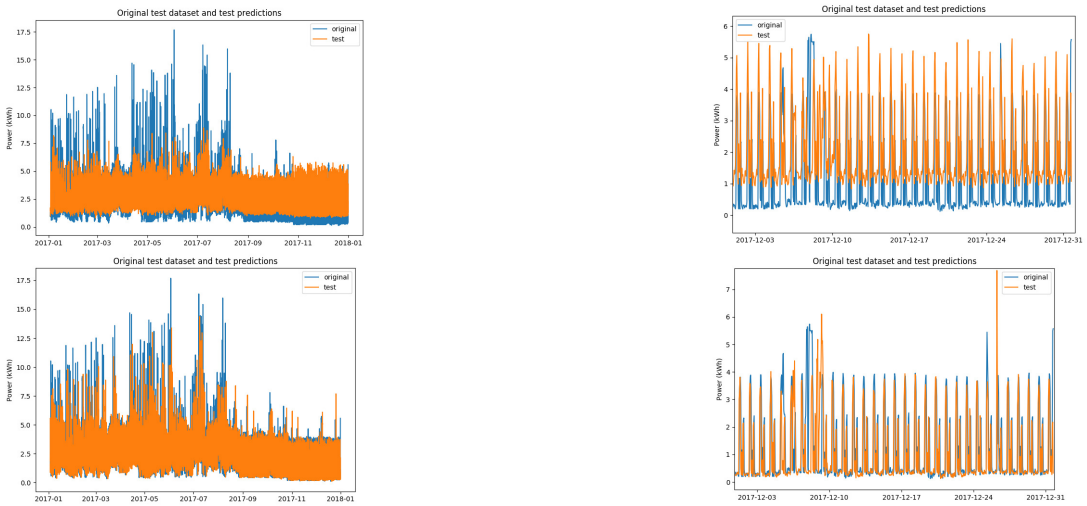


Figure 2: Forecasts for the entire test dataset are presented on the left images for both CapsNet (top) and CNN (bottom). Results for the last month are also presented on the right, for CapsNet (top) and CNN (bottom).

During training, CapsNet registered RMSE of 1.394, MAE of 0.626 and MAPE of 22.037%, against RMSE of 1.158, MAE of 0.563 and MAPE of 21.878% for the CNN. Higher error values were recorded for the test dataset, with a RMSE of 1.504, MAE of 0.739 and MAPE of 32.797% for CapsNet, and RMSE of 1.222, MAE of 0.609 and MAPE of 26.542% for CNN.

6 Conclusion

In this work we studied the application of capsule networks to forecast domestic energy consumptions. Capsule networks are a novel technique, proposed to address problems of CNNs in image classification and computer vision. To the best of our knowledge, these networks were only sporadically applied for time-series forecasting.

Capsule networks appear as interesting and promising alternatives to CNNs, since they remain capable of exploring spatial relationships between input data features that CNNs can't: besides downsampling the feature size, CNN's max pooling only keeps information about the presence of a given feature, discarding its location in the input.

Indeed, given a window of previous values, forecasts may be substantially different if a given pattern is observed at the start rather than at the end (therefore, closer to the present and the next estimation) of the window. For this reason, we consider that capsule networks can make important contributions to improve computed forecasts.

Our CapsNet architecture was not able to capture time-series dynamics, computing poor forecasts. Although an high error was still obtained, the CNN model provides better approximations. CapsNet's predictions are somewhat stationary, almost as if confined to a given range. CNN model, on the other hand, is better at following the shape of the series. From these results, it is clear that further work on these architectures must be carried out in order to improve their forecasts.

Acknowledgement

Joaquim Leitão gratefully acknowledges the Portuguese funding institution FCT – Foundation for Science and Technology –, Human Capital Operational Program (POCH) and the European Union (EU) for supporting this research work under the Ph.D. grant SFRH/BD/122103/2016.

References

[1] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[2] Mehreen S Gul and Sandhya Patidar. Understanding the energy consumption and occupancy of a multi-purpose academic building. *Energy and Buildings*, 87:155–165, 2015.

[3] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[4] Pecan Street Inc. Pecan street dataport. `https://dataport.cloud/`, 2018. [Online; accessed August 10, 2018].

[5] Youngjoo Kim, Peng Wang, Yifei Zhu, and Lyudmila Mihaylova. A capsule network for traffic speed prediction in complex road networks. *arXiv preprint arXiv:1807.10603*, 2018.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Joaquim Leitão, Paulo Gil, Bernardete Ribeiro, and Alberto Cardoso. Application of bees algorithm to reduce household's energy costs via load deferment. In *Proceedings of the 10th International Conference on Information Technology and Electrical Engineering*, 2018.

[8] Joaquim Leitão, Paulo Gil, Bernardete Ribeiro, and Alberto Cardoso. Improving household's efficiency via scheduling of water and energy appliances. In *Proceedings of the 13th APCA International Conference on Automatic Control and Soft Computing*, 2018.

[9] Krist Papadopoulos LLC. Deep learning forecasting with dilated convolutional neural networks on the cif2016 dataset, 2018. URL `https://github.com/kristpapadopoulos/seriesnet/blob/master/SeriesNet-Krist_Papadopoulos.pdf`.

[10] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[11] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

Paper:#19

Sheet Music Player based in Image Processing

By: Caridade, C. Rosendo, S.

Sheet Music Player based in Image Processing

Cristina M.R. Caridade
caridade@isec.pt
Sara Rosendo
rsara055@gmail.com

Department of Mathematics and Physics
Coimbra Polytechnic, ISEC
Coimbra, Portugal

Abstract

The aim of this paper is to present an automatic application developed to detect musical symbols in digital sheet images and their reproduction. The application was developed in Matlab using digital image processing techniques. For this purpose, the application used converts the image to gray and then to binary image where the stave and lines are located. Stave are separate and lines are deleted. Then all the notes of each stave are recognized by their signature and beat. Finally the sound was reproduced.

1 Introduction

The passing of the years and the consequent degradation of old documents leads to loss of great historical value, which is reflected in the need to act in a way to conserve this information. As its manual preservation is very expensive and time consuming, a more effective alternative is to digitize the documents [1]. However, the scanning process only allows the contents of documents to be saved, it is important to create a system that performs character recognition and reproduces them [2-4]. There are already commercial systems available in market. Some quite expensive, others cheaper, however the detection and recognize of musical sheets scanned or in photograph is not yet completed, by the number of actual papers that refer to this subject [5-7].

This paper presents an automatic application developed in Matlab that reproduces music written in a sheet automatically. The development of this application is very important, since it not only contributes to a greater and easier preservation of musical sheets, but also that this type of music reaches a greater number of people who do not know enough to reproduce it in another way.

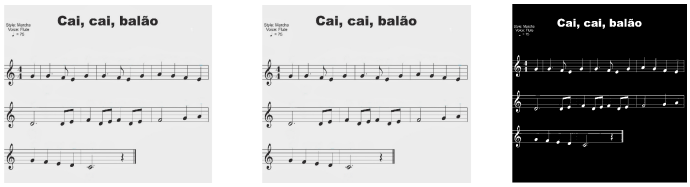
2 Methodology

The application is defined by three main steps: Image preprocessing, image segmentation and note detection and reproduction.

2.1 Image preprocessing

2.1.1 Binary Image

The first step is to transform the original image (music sheet) into a binary image. The original image is initially converted to a grayscale image and finally to a binary image, using the Otsu method, as shown in Figure 1.

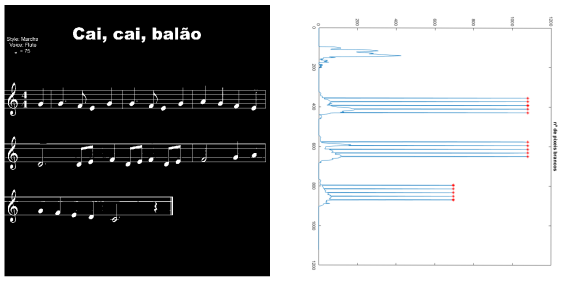


(A1) (A2) (A3)

Figure 1: Original (A1), grayscale (A2) and binary image (A3).

2.1.2 Image histogram

After obtaining the binary image its histogram by lines is calculated to detect the stave location in the image. In Figure 2 we can see the binary image $A3$ and its histogram by lines. As can be seen the peaks (represented in red) of the histogram (maximum numbers of white pixels) represent the 5 lines of each stave. In this case the image has 3 staves.

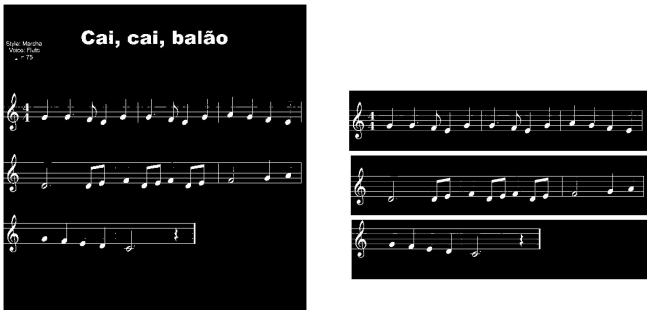


(A3) (Histogram)

Figure 2: A3 image and his histogram by lines.

2.1.3 Staves separation from the image

Now it is necessary to separate the staves in the image. For this, the distance in x between the consecutive maximum peaks and the intervals where this distance is greater (the intervals between staves) were calculated. In Figure 3 the $A3$ image is subdivided into 3 sub-images that correspond to the 3 staves ($A3a$,$A3b$ and $A3c$).



A3 A3a (up), A3b (middle), A3c (down)

Figure 3: Staves separation from A3 image.

2.2 Image segmentation

2.2.1 Lines and objects elimination

For each stave the lines are eliminated by subtract to the binary image, a mask defined by a black image with the same dimension with the lines defined in white. This process is important because it allows us to remove the white pixels that define the lines and that make difficult the correct identification of musical notes and symbols. In Figure 4 it is possible to observe the lines elimination of the stave $A3a$. In this image only the musical notes and symbols that constitute the stave are represented. To remove the clef and time signature at the beginning of each stave it is necessary to find all the objects and apply a mask to eliminate them. Figure 4 shows the $A3a_2$ image obtained from the image $A3a_1$ by eliminating the initial objects (clef and time signature).

$A3a_1$ $A3a_2$

Figure 4: A3a stave without lines (left) and objects (right).

2.2.2 Deleting objects at the edges and filling holes

First, objects that touch the edges are eliminated. Than, a 3-pixel diameter disk element is used and the morphological operator close to fill existing holes. The result of this operation on the image $A3a_2$ is represented in the image $A3a_3$, shown in figure 5.

2.2.3 Vertical Lines elimination and center of mass

To eliminate the barlines and the stem of the musical notes, a structuring element in the form of a line with a length of 10 pixels and an angle of 180° is used as operator in the morphological erosion application. For objects whose area is within a range of 20% of the average area of all objects the center of mass (centroid) is calculated. In the image $A3a_4$ of figure 5 the center of mass of the notes are represented by an asterisk in the correct spatial location.

$A3a_3$ $A3a_4$

Figure 5: A3a without objects at edges and with filling note holes (left) and center of mass (right).

2.3 Note detection and reproduction

2.3.1 Note signature

To calculate the music note signature of each note, it is necessary to identify where the note is in relation to the 5 lines of the stave. For this, the values of y of the center of mass of each note will be compared with the values of y of the 5 lines. Figure 7 shows the signature of each musical note through the image A3a. The first note, for example, is "Sol" because it is located on the second line and the third one that is between the first and the second line is called "Fa".

2.4 Note beat

To calculate the note value of each note, it is necessary to identify how each note is made up. A simple note (Figure 6) is formed by a stem, a flag and a note head, with or without a point in the left side.

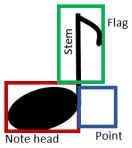


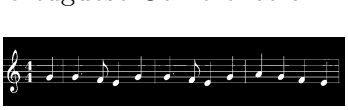
Figure 6: Musical note constitution.

- Holes - To check if the note head has a hole (Holes), a box is created around the head note, and was check if there are black pixels inside. If the center pixels are equal to zero (black), it means that the head of the note is not filled, that there is Hole and is assigned to the note value 2 more beats.

- Point - To check if the note contains a point (Point) on the right, a box is defined to the right of the hole, as shown in figure 6 in blue. Within the region defined by this box are detected their objects. A local analysis of existing objects and their eccentricity is done. For this, a dilatation with a structuring element in the form of a circle is applied and the eccentricities of all the objects found are calculated. If there is a point, it is detected as an object with eccentricity close to the eccentricity of the circle and the note is assigned 2 more beats.

- Flag - For the identification of the flag type of the note (Flag) a box is defined that surrounds it (green box in figure 6). A local analysis is made to the previously defined region and a morphologic operation dilation is applied to join the lines defining the Flag. Then the objects present and their orientation are detected. If there are non-vertical objects, they define the tail type of "eighth notes" and therefore the corresponding note value will have -2 beats.

The note value of the each of the 13 notes defined in the A3a image initially is 4 beats. The second note will have +2 beats (Point), the third -2 beats (Flag), the sixth +2 beats (Point) and the seventh -2 beats (Flag). Thus the note value assigned to these 13 notes is 4, 6, 2, 4, 4, 6, 2, 4, 4, 4, 4, 4. Figure 7 shows the $A3a$ image with the respective note beats below.



note signature: Sol, Sol, Fa, Mi, Sol, Sol, Fa, Mi, Sol, La, Sol, Fa, Mi

note beat: 4, 6, 2, 4, 4, 6, 2, 4, 4, 4, 4, 4

Figure 7: Note signature and beat of $A3_a$ image.

2.5 Sound reproduction

Finally after the detection of all the notes (signature and beat) the sound was reproduced in Matlab. Each signature is associated with a unique frequency [8]. With the information of the signature and beat, it is possible to generate a sinusoidal wave that represent the note. Than, it is necessary to concatenate the sinusoidal waves for each note together, and output the sound of the resulted wave.

2.6 Experiments

The application developed was tested on 10 simple music scores as shown in Figure 1 and 8. The staves separation rate (number of detected staves divide by total number of staves) is 100%. The holes head, flag and point detection rate (number of detected holes, flag, point divide by total number of note holes, flags and points) is about 98% in each case. For the other types of music scores the application can not be used because in its development it was made some assumptions that will be fix in the future.

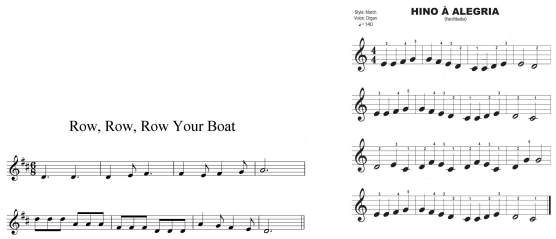


Figure 8: Some other music sheets.

3 Conclusion and future work

Arranging ways to play digital sheets is a relevant area that has been explored. The elaboration of this work showed that the segmentation and the correct choice of values for each stage are fundamental parts for a successful result, just as important is the perception of each phase and of the methodologies to follow.

One of the future objectives is to improve the process of recognizing and reproducing musical notes in order to be able to extend the result to sheets with more prompts and with a greater degree of complexity, as well as to allow the recognition of images of handwritten sheets [9].

4 References

[1] T. Pinto, A. Rebelo, G. A. Giraldi, J. S. Cardoso. Music score binarization based on domain knowledge. In Proceedings of the 5th Iberian Conference on Pattern. Recognition and Image Analysis. Las Palmas de Gran Canaria, Spain, 700-708, 2011.

[2] A. Rebelo, J. S. Cardoso. Staff line detection and removal in the grayscale domain. 2013.

[3] C. Solomon, T. Breckon. Fundamentals of Digital Image Processing: A Practical Approach Using Matlab. Wiley 2011.

[4] F. Tajeripour, M. Sotoodeh. A novel staff removal method for printed music image. *IEICE Electronics Express*, 9 (7):609-615. 2012.

[5] J. Calvo-Zaragoza, G. Viglensoni, I. Fujinaga. Pixel-wise binarization of musical documents with convolutional neural networks. In Fifteenth IAPR International Conference on Machine Vision Applications, 362-365, 2017.

[6] J. Calvo-Zaragoza,D. Rizo. Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In Proceedings of the 19th International Society for Music Information Retrieval. Paris, France, 2018.

[7] A. Pacha, J. Calvo-Zaragoza. Optical Music Recognition in Mensural Notation with Region-Based Convolutional Neural Networks. In Proceedings of the 19th International Society for Music Information Retrieval. Paris, France, 2018.

[8] Piano Key frequencies, Wikipedia,
https://en.wikipedia.org/wiki/Piano_key_frequencies

[9] A. Castro. Reconhecimento de símbolos musicais em imagens cinza de partituras manuscritas. *Tese de Mestrado FEUP*. 2014.

Paper:#20

Locally Affine Light Fields as Direct Measurements of Depth

By: Marto, S. Monteiro, N. Gaspar, J.

Locally Affine Light Fields as Direct Measurements of Depth

Simão Pedro da Graça Oliveira Marto
smarto@isr.tecnico.ulisboa.pt
Nuno Barroso Monteiro
nmonteiro@isr.tecnico.ulisboa.pt
José António Gaspar
jag@isr.tecnico.ulisboa.pt

Institute for Systems and Robotics
Instituto Superior Técnico
University of Lisbon, Portugal

Abstract

Light field imaging allows discriminating object radiance according to multiple viewing directions. We introduce the minimal light field representation from which depth can be extracted, the affine light field, which is a first order approximation. One setup to acquire one globally affine light field is proposed. Consequently, we show how Dansereau Bruton's gradient based reconstruction method [1] can be derived from the locally affine light field assumption.

1 Introduction

Light field cameras, sometimes called plenoptic cameras, have been introduced recently to the consumer market [4]. They are capable of discriminating the contribution of each light ray emanating from a particular point by projecting the point to several positions of the sensor.

A light field image is usually represented by the 4D plenoptic function [4], but it can be seen as a collection of 2D viewpoint images, each with a projection center slightly offset (details in section 2). This means that from a light field image it is possible to extract depth information. The only requirements is that the gradients in a viewpoint image are not null.

In this paper we introduce a minimal order approximation for a light field image which still contains depth information. It is a first order approximation due to the constraint that the gradients cannot be null. We refer to such light fields as globally or locally affine. An example setup to capture a globally affine light field is illustrated in Fig. 1. We use this approximation to derive the formula to extract depth from a light field image.

2 Light Field Camera Model

A light field image is a mapping of rays into light intensities. We make the distinction between the light field in the object space, indexed by rays in the object space, and the light field in the image space, indexed by rays in the image space. When a light field image is captured, it's in the image space, but in order to obtain metric information about a scene, the light field must first be converted into the object space.

The model proposed by Dansereau *et al.* considers a mapping between rays in the image space, sometimes referred to as raxels [4], and rays in the object space. This is the light field equivalent of an intrinsic camera model.

The rays in the object space are modelled with the two plane parameterization, see Fig. 2. Each ray is defined by its intersection with a plane (s,t) and its direction is defined by slopes (u,v) relative to the z axis. To help illustrate this parameterization, and to facilitate the understanding of the calculations in the next sections, one writes $(x,y) = (s,t) + z \cdot (u,v)$ to show how the (x,y) coordinates of a point along a ray (s,t,u,v) can be calculated from its z coordinate.

The typical construction of a light field camera is based on an array of microlenses placed between the camera main lens and the imaging sensor (usually a CMOS). The raw image extracted from the CMOS results in the so-called image in the image space after a decoding process.

In the image space, coordinates (k,l) indicate the microlens the ray passed through before sampling, and (i,j) indicate the pixel within the microlens image. Alternatively, (i,j) can be seen as selecting a viewpoint, and (k,l) as selecting a pixel within that viewpoint image. Changing (i,j) changes the projection center of the viewpoint image slightly within a plane parallel to the image plane. Another useful construct is the Epipolar Plane Image (EPI), obtained by fixing (j,l) (horizontal EPI) or fixing (i,k) (vertical EPI).

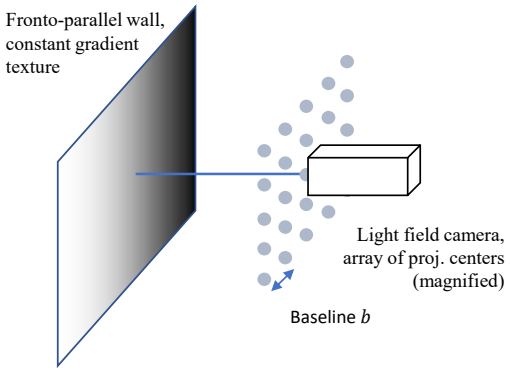


Figure 1: Setup to acquire a globally affine light field with a light field camera. The central circle represents the projection center of the central viewpoint of the light field camera. The array of circles represents the array of projection centers (not in scale) representing the other viewpoints of the light field camera.

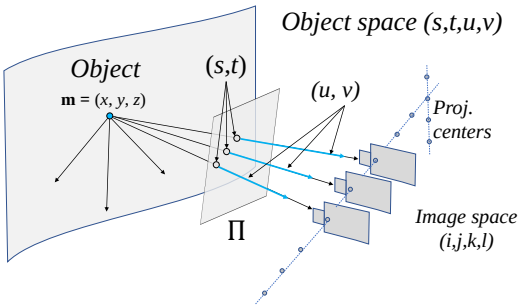


Figure 2: Two plane parameterization for rays starting from a point **m** using a point and a direction. The point (s,t) is given by the intersection with the plane Π, and the direction (u,v) with the derivative of the ray's coordinates with respect to z. The latter coordinates can also be seen as the intersection with a plane perpendicular to the first at a distance of one unit, hence the name "two plane parametrization".

The model defined in [2] takes the form of:

$$\underbrace{\begin{bmatrix} s \\ t \\ u \\ v \\ 1 \end{bmatrix}}_{\Psi} = \underbrace{\begin{bmatrix} h_{si} & 0 & h_{sk} & 0 & h_s \\ 0 & h_{tj} & 0 & h_{tl} & h_t \\ h_{ui} & 0 & h_{uk} & 0 & h_u \\ 0 & h_{vj} & 0 & h_{vl} & h_v \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} i \\ j \\ k \\ l \\ 1 \end{bmatrix}}_{\Phi} . \tag{1}$$

The model in Eq. 1 has 8 parameters. The values in the last column of the matrix are not independent parameters, they are set by the requirement that Ψ_{center} should map to Φ_{center}. With a simple and reasonable set of assumptions, these can be reduced to just two parameters, and a meaning can be assigned to them by making an analogy to a camera array.

The first simplifying approximation is to consider the parameters referring to the horizontal and to the vertical coordinates to be equal. This is supported by the fact that, although the microlens array structure is hexagonal, a decoding algorithm can re-sample the microlenses in a square lattice, as is done by Dansereau *et al.* in [2].

Afterwards, we can move the (s,t) plane along z, such that it now includes the centres of projection of the viewpoints. The result is a new $\mathbf{H}_a$ matrix that describes the exact same camera, but has $h_{sk} = h_{tl} = 0$.

The translation in z would be compensated by an opposite translation in the extrinsic parameters.

Furthermore, the terms h_{ui} and h_{vj} describe a shift of the principal point of each viewpoint image proportional to (i, j). Since this shift can be easily removed from an image, we will consider it to be zero.

Combining all of these simplifications, we get an intrinsics matrix $\mathbf{H}_a$ with 2 parameters (apart from the 4 in the last column, which continue to be set by the requirement that Φ_{center} maps to Ψ_{center})

$$H = \begin{bmatrix} b & 0 & 0 & 0 & s_0 \\ 0 & b & 0 & 0 & t_0 \\ 0 & 0 & f^-1 & 0 & -c_x/f \\ 0 & 0 & 0 & f^-1 & -c_x/f \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad , \tag{2}$$

where the parameters $h_{si} = h_{tj}$ and $h_{uk} = h_{vl}$ are replaced by b and f^{-1}. The reason for this substitution is that these terms now have meaning in terms of a camera array: b is the baseline, or the distance between adjacent cameras; f is the focal length of the cameras. A more detailed explanation of the intrinsics matrix applied to a camera array can be found in [3].

3 Affine Light Field and Depth Estimation

The light field of a fronto-parallel plane *colored* with a gradient, Fig. 1, is the simplest scenario producing an affine light field. To show this, consider a fronto-parallel plane Π where $\mathbf{n} = (0,0,1)$, and $r = z$, such that $\mathbf{p} \in \Pi \implies \mathbf{p} \cdot \mathbf{n} = r$. The color of the plane at a point $\mathbf{p}$ in the plane Π, is given by $c(\mathbf{p}) = \mathbf{p} \cdot \mathbf{g} + c_0$, where $\mathbf{g}$ is the color gradient, and is a vector aligned with the plane.

To find out the color sampled by a ray Ψ, we find out where it hits the plane Π using the back projection equation $\left([s\ t\ 0]^T + \lambda [u\ v\ 1]^T\right) \cdot \mathbf{n} = r$. Note that λ, the parameter representing how much the ray extends before hitting Π, is actually the depth z of this plane at that point. From here on, we will use $z = \lambda$. Hence one has $z = \dfrac{r - (s,t,0) \cdot \mathbf{n}}{(u,v,1) \cdot \mathbf{n}} = r$. Combining with the camera parameterization in Eq. 1, we get the affine light field:

$$L(i,j,k,l) = l_0 + \begin{bmatrix} a_i\ a_j\ a_k\ a_l \end{bmatrix} \cdot [i\ j\ k\ l]^T \quad , \tag{3}$$

where $a_i = bg_x$, $a_j = bg_y$, $a_k = zg_x/f$ and $a_l = zg_y/f$ and l_0 collects all the constant terms. The gradient of L, $\nabla L = \begin{bmatrix} a_i\ a_j\ a_k\ a_l \end{bmatrix}^T$, contains the depth z only in the (k,l) derivatives. The only other unknown parameters, g_x and g_y, are present in both (i, j) and (k,l) and so can be cancelled by dividing a_k with a_i or a_l with a_j. Hence, the affine light field produces directly a depth estimate

$$z = bf\frac{a_k}{a_i} \quad \text{and/or} \quad z = bf\frac{a_l}{a_j} \quad . \tag{4}$$

Comparing Eq. 4 with stereo reconstruction, one finds, similarly, the baseline and focal length, while a_i/a_k and a_j/a_l do the role of disparities.

In order to use Eq. 4 to extract depth from a real scene, one has to estimate the values of $a_{(\cdot)}$, by calculating a locally affine approximation. This can be done by estimating the gradients in the EPI's, based on Sobel operators, as in [1]. Alternatively, in [5] the structure tensor is used, which involves derivative estimates in the four components of the light field combined with low pass filtering in the four dimensions, in order to attenuate high frequency noise enhanced by the derivative operations. We use the structure tensor formulation. When both a_i and a_j are not zero we output the mean of the two z estimates from Eq. 4. If just one value is not zero then the depth estimate is based just on that value.

4 Experiments and Results

In a first experiment a synthetic figure is created following the setup in Fig. 1. The camera parameters in our experiment are $b = 3 \times 10^{-4} m$ and $f = 200$, while the parameters of the scene are given by $\mathbf{g} = (1,0)m^{-1}$ and $z = 0.15m$, which theoretically results in a light field given by $L = 10^{-4} i + 7.5 \times 10^{-4} k$. From the resulting lightfield, the gradients can be extracted using the structure tensor as in [5] without the regularization step. Even in this simple setup, one has to contend with errors induced by quantization of the image signal, in our case 8 bits. Nonetheless, the reconstruction returned robust results of $z = 0.149 \pm 0.008\ m$.

In a second experiment we considered a more involved setting, a spherical hubcap on top of a plane with a gradient, as represented in Fig. 3. In this case the light field is not globally affine on the hubcap. The same reconstruction method was applied with the results illustrated in Fig. 4. Good results were obtained even on non globally affine light fields, since they are still locally affine, i.e. are well represented locally by a first order approximation. The mean of the absolute relative errors obtained was 1.49%.

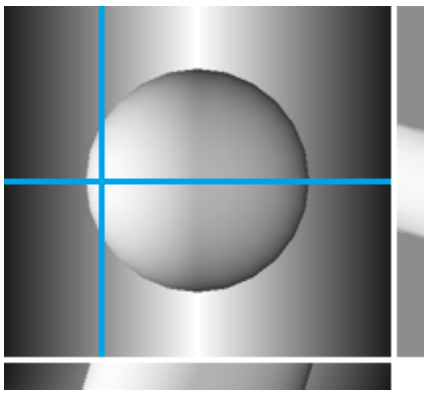


Figure 3: Example light field image to demonstrate depth reconstruction. Central viewpoint surrounded by two EPI's. The bottom and right EPI's originate from the horizontal and vertical lines, respectively.

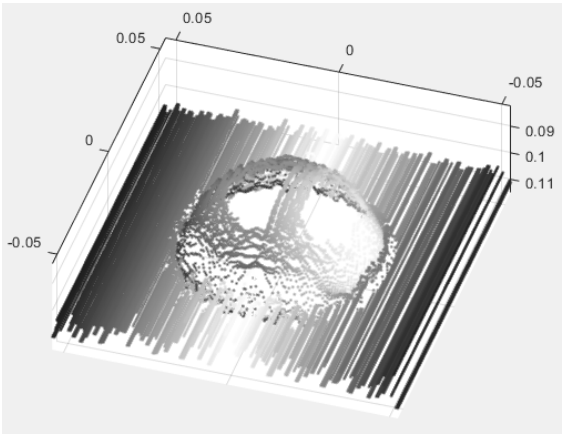


Figure 4: Reconstruction of the synthetic light field image. Depth values are measured with respect to the camera coordinates frame.

5 Conclusions

In this paper we have shown how a light field camera model and its produced images can be interpreted in familiar terms, so as to facilitate the reconstruction of the 3D objects captured. Furthermore, we introduced a minimal order light-field containing depth information which can be extracted by light-field analysis

Acknowledgements

Work partially supported by the FCT project UID / EEA / 50009 / 2013.

References

[1] D. Dansereau and L. Bruton. Gradient-based depth estimation from 4d light fields. In *IEEE ISCAS*, volume 3, pages III–549, 2004.

[2] D. Dansereau, O. Pizarro, and S. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *IEEE CVPR*, pages 1027–1034, 2013.

[3] S. Marto, N. Monteiro, J. Barreto, and J. Gaspar. Structure from plenoptic imaging. In *IEEE ICDL-EpiRob*, volume 18, 2017.

[4] Ren Ng. *Digital light field photography*. PhD thesis, stanford university, 2006.

[5] S. Wanner, C. Straehle, and B. Goldluecke. Globally consistent multi-label assignment on the ray space of 4d light fields. In *IEEE CVPR*, pages 1011–1018, 2013.

Paper:#25

Application of Lifelong Learning with CNNs to Visual Robotic Classification Tasks

By: Zacarias, A. Alexandre, L.

Application of Lifelong Learning with CNNs to Visual Robotic Classification Tasks

Abel S. Zacarias
http://www.di.ubi.pt
Luís A. Alexandre
http://http://www.di.ubi.pt/~lfbaa/

Departamento de Informática
Universidade da Beira Interior
6201-001 Covilhã, Portugal
Instituto de Telecomunicações

Abstract

The field of robotics is becoming continuously more important, due to the impact it can bring to our everyday life. A long standing problem with neural network learning is the catastrophic forgetting when one tries to use the same network to learn more than one task. In this paper we present results of the application of a method to avoid catastrophic forgetting while using Convolutional Neural Networks (CNNs) to some visual recognition tasks relevant to the field of robotics. The results show that with this method a robot can learn new tasks without forgetting the previous learned tasks. Results also showed that if we applied this method, the performance on isolated tasks increases and it is better to use it than train a CNN in an isolated way (single task). We use for our experiments two well known data sets, namely, Olivetti Faces and Fashion-MNIST.

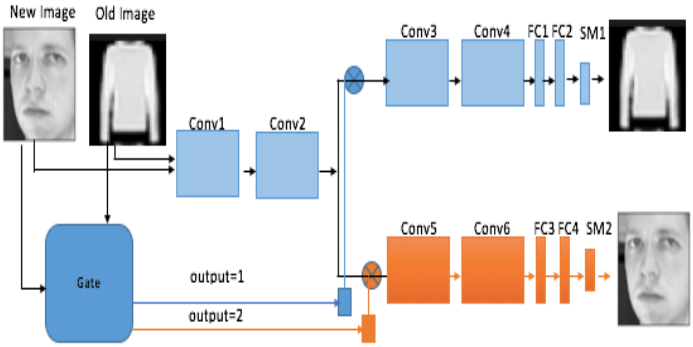


Figure 1: Network architecture: one branch is used for each task and the gate selects the branch to deploy at test time.

1 Introduction

Object recognition is an area of computer vision that deals with ability of an intelligent system to recognize instances of objects belonging to a certain category. This task has been important to the scientific community because of its many applications. One of these applications is in robotics, where a robot learns to distinguish objects (eg. boats, cars, clothes, plates, etc.) such that it can interact with the world. Another important computer vision area is biometrics, where a system processes data to recognize persons, such as using face images. Both these tasks are important if one wants to have robots interacting with people. Currently, deep learning approaches, such as the use of Convolutional Neural Networks (CNNs), has shown to be very effective in these types of visual recognition tasks. One goal is to have a single system learning to solve several tasks by reusing information from previously learned tasks to improve new ones, without forgetting what was learned before. This forgetting, in the neural network context, is a problem called catastrophic forgetting [2]. Catastrophic forgetting means that the performance of an agent trained to recognize a given task decreases when new tasks are added in an incremental manner.

There are some approaches to mitigate or overcome this the problem of catastrophic forgetting. For example [1] proposed solving the catastrophic forgetting problem in incremental learning scenario with support data inspired by the two major neurophysiology theories (Hebbian Learning System and complementary learning system). Also, in [6] an approach based on reinforcement learning is proposed where an agent interacts with an unknown environment to solve a specific task according to a policy and reward signal. It consists of three networks: controller, value network and task network. After learning the first task, the controller network decides how many filters or nodes should be added to each layer corresponding the new tasks, and only train the network on new added filters or nodes. The task network correspond to the expanded child network obtained from the controller network. During training, the parameters from the first task were frozen and only back-propagated the new added nodes or layers. Here we apply the SENA-CNN approach [7] to avoid catastrophic forgetting problem where the agent can incrementally learn new tasks without forgetting what was previous learned, while working with CNNs, and apply it to learning two problems relevant in the context of robotics: object and face recognition, using a single network.

2 Proposed Method

In this paper we aim to apply the method for lifelong learning proposed in [7] to the field of robotics. Our idea is to start by teaching a robot how to recognize faces and progressively teach it more capabilities in such a way that as it learns how to solve new tasks it does not forget the ones it previously learned.

First, the system must learn how to do face recognition, as it is a fundamental task in human-robot interaction; second, we will teach it to do object recognition, as it is also a key capability for a robot to be able to interact with its environment. To do so, we begin by training a CNN in one task. After the model reaches convergence, we make it learn the second task. For each new task, we add a new branch to the first model trained in an isolated way. For the second task, we only train our model on this added branch and keep the parameters of the previous layer frozen.

We do not add the first two layers of the model corresponding to the second task, instead, we use the two layers of the first learned task. That's because the neurons in those layers find several simple structures, such as oriented edges as demonstrated in [3] and these are applicable to many different tasks. The remaining layers seem to be devoted to more complex representations, and hence, are more specific to each problem, and that is why we choose to create those new layers instead of re-using the original ones.

2.1 Using the Gate to Select the Correct Branch

Our goal is to train a gate network with a super-class of both tasks. Each task is labelled $\{0,1,...,n-1\}$, where n represents the total number of tasks to be deployed. This way, there is no need for all branches to process the input and produce an output and hence, only one branch is chosen by the gate to make the final decision. Figure 1 shows the procedure used by the gate mechanism for the case we are solving in this paper ($n = 2$).

2.2 Training Methodology

The network is going to learn how to solve several tasks. First the gate is trained by using images from each task, with labels that represent those tasks (and not the original task classes). Second, we train a first network branch, starting with randomly initialized weights. Then, for each new task, a new branch is added to the network, where the first two convolutional layers are reused from the first branch.

3 Experiments

In this section we present the results of our method applied to the robotics field. We conducted our experiment using two data sets namely, Fashion-MNIST [5] which consists of 28×28 grey-scale images with 60000 training set and 10000 test set. We also used for our experiments the Olivetti data set [4] which comprises 400 pictures of people, 10 per person. All images are in grey-scale with size 64×64, all frontal and with a slight tilt of the head.

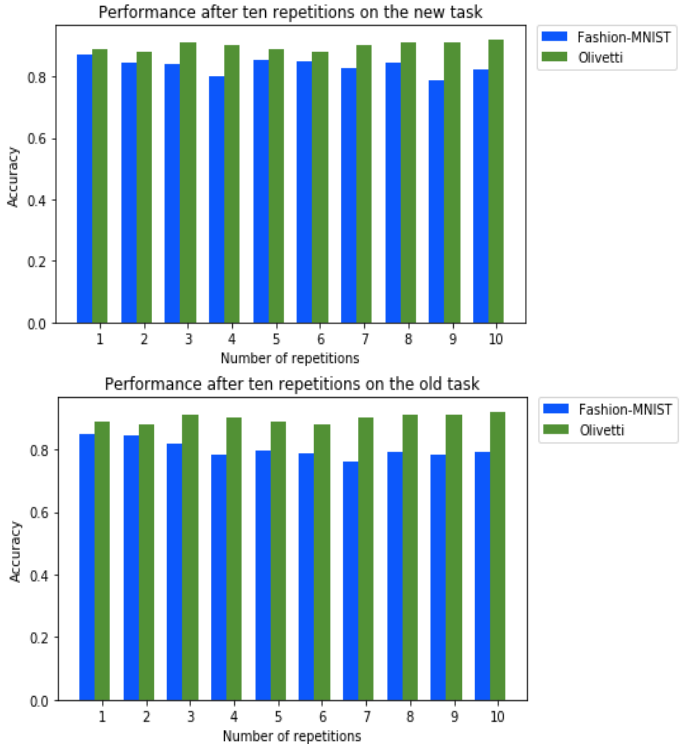


Figure 2: Network performance when we add a new task to an existing network trained on isolated learning.These results correspond to the ten repetitions of train and test of the proposed model. After testing the new task we test again on the old task: (top) performance on new added task; (bottom) performance on old task after training the network with the new task.

3.1 Network Architecture

To run our experiments we used a convolutional neural network with four convolutional layers each with a ReLU activation layer, maxpooling layer, dropout, flatten layer and two dense layers. The last dense layer is connected with a softmax layer. The gate network has the same architecture as the CNNs used in each branch.

As we previously said, the gate network is used to choose which branch to deploy at test time. Table 1 presents the mean accuracy and (standard deviation) when training each task in isolated learning, after ten repetition. These two networks trained on isolated are then used to add the branch corresponding to each new task.

Train	Test	Baseline
Gate	Gate	99.29 (2.17)
Olivetti	Olivetti	94.00 (3.25)
Fashion-MNIST	Fashion-MNIST	77.55 (11.35)

Table 1: Performance results on isolated learning of the gate, Olivetti and fashion-MNIST.

3.2 Adding New Tasks to the Model

As previously said, after training the robot to perform the first task, is necessary to teach the robot to perform new tasks as they come sequentially with the ability of not forgetting what was learned before. A robot with this ability will be a good improvement in the field of artificial intelligence, because the robot can learn many tasks without forgetting over its life time.

Table 2, presents the mean accuracy (and standard deviation) of our method when learning the new task. As we can see in the table, results show that if we see the two layers of the model trained on isolated learning to learn a new task, it can increase the performance of the new task compared to using a model trained isolated on the correspondent task. In this case, the performance of a robot will increase if we use this approach to learn new tasks.

After learning the new task it is necessary to test if the model did not forget what it learned previously. Table 3 presents the performance of our model after learning the new task. The Table shows that our model was able to preserve the performance on the old learned task and so it was possible for the robot to learn a new task without forgetting what it learned previously.

Old	New	Accuracy
Fashion-MNIST	Olivetti	94.41 (2.02)
Olivetti	Fashion-MNIST	83.23 (2.39)

Table 2: Network performance on new task, after learning a first task.

New	Old	Accuracy
Fashion-MNIST	Olivetti	91.43 (2.06)
Olivetti	Fashion-MNIST	79.50 (2.71)

Table 3: Network performance on old task, after learning a new task.

Verifying the experiments results, it is possible to observe the ability of the proposed approach to deal with the catastrophic forgetting problem. Comparing the results of a model trained on isolated learning with the proposed method, there is a slight degradation of performance on old task for Olivetti data set that is less than 3%. Another interesting observation is that the proposed method showed in the two scenarios (Fashion-MNIST $\mapsto$ Olivetti and Olivetti $\mapsto$ Fashion-MNIST) increased performance when compared to isolated training. This is understandable since by reusing partial information from previous tasks, we are somehow doing fine-tuning on the new task.

4 Conclusion

In this paper we present the results of the application of lifelong learning with CNNs to object and face recognition tasks. The results showed that we can use the SENA-CNN model to perform lifelong learning in this context and a robot using this approach is more efficient than if it is trained to learn each task with independent CNNs. Next we will focus on using theses results in a real scenario with a robot.

5 Acknowledgment

This work was supported by National Founding from the FCT- Fundação para a Ciência e a Tecnologia, through the UID/EEA/50008/2013 Project. The GTX Titan X used in this research was donated by the NVIDIA Corporation.

References

[1] Y. Li, Z. Li, L. Ding, Y. Pan, C. Huang, Y. Hu, W. Chen, and X. Gao. SupportNet: solving catastrophic forgetting in class incremental learning with support data. *ArXiv e-prints*, June 2018.

[2] Zhizhong Li and Derek Hoiem. Learning without forgetting. *CoRR*, abs/1606.09282, 2016. URL `http://arxiv.org/abs/1606.09282`.

[3] Ivet Rafegas, María Vanrell, and Luís A. Alexandre. Understanding trained cnns by indexing neuron selectivity. *CoRR*, abs/1702.00382, 2017. URL `http://arxiv.org/abs/1702.00382`.

[4] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142, Dec 1994. doi: 10.1109/ACV.1994.341300.

[5] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL `http://arxiv.org/abs/1708.07747`.

[6] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *CoRR*, abs/1805.12369, 2018. URL `http://arxiv.org/abs/1805.12369`.

[7] A. Zacarias and L.A. Alexandre. Improving sena-cnn by automating task recognition. In *19th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, LNCS, Spain, Madrid, 21-23 November 2018. Springer.

Paper:#38

Comparing Learning Approaches for Twitter Sentiment Analysis

By: Guevara, J. Morales, M. Costa, J. Silva, C.

Comparing Learning Approaches for Twitter Sentiment Analysis

Juan Guevara[1,3]
2162315@my.ipleiria.pt
Mario Morales[3,4]
mmorales@grupo-novatech.com
Joana Costa[1,2]
joana.costa@ipleiria.pt
Catarina Silva[1,2]
catarina@ipleiria.pt

[1]School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2]Center for Informatics and Systems of the University of Coimbra, Portugal
[3]Universidad Central del Ecuador, Ecuador
[4]University of Alicante, España

Abstract

Nowadays, sentiment analysis is a popular technique for the analysis of social networks. One of the most popular social networks and microblogs is Twitter, which allows a user to express his/her opinions using short texts. That is why it is important to analyze and know what people think about a topic and finally provide help to make decisions in the face of some problem. In this work we compare different machine learning approaches for sentiment analysis in Twitter social network.

1 Introduction

In recent years, sentiment analysis has been used in many areas and by many companies. Within the analysis we can have social networks, as well as one of the main microblogging, such as Twitter. In this social network a lot of information given by users is generated every day, since they can freely express themselves on some topic and share with more users. It is for this reason that this microblogging has had a lot of impact on current society and is a key piece to know what people think about a certain problem. To analyze all this information, we can make use of machine learning techniques to automate the analysis and finally obtain results.

The aim of this work is to compare the learning techniques for sentiment analysis in Twitter social network, in order to determine the behavior of each of them. The structure of the paper is as follows. Section 2 shows the background on sentiment analysis. In the Section 3 we define the proposal and the work to be carried out. Section 4 shows the configuration that was made for the sentiment analysis with each of the techniques, Section 5 presents the test results and the analysis, and finally in Section 6 we have the conclusions and future work.

2 Background

Sentiment Analysis (SA) is based on the idea of processing or analyzing opinions in an automatic way to obtain a fundamental value and adequate decision making, through Natural Language Processing (NLP) [1].

Text classification methods that use the machine learning approaches can be roughly divided into: supervised and unsupervised learning methods. The supervised methods make use of a large number of previously labeled documents and unsupervised methods are used when it is difficult to find these documents already labeled [2].

2.1. Machine learning

The machine learning approach is based on the use of algorithms that allow learning to computers, through prior knowledge; In our work the problem will be for the classification of tweets through the use of these algorithms.

For the definition of the problem of SA in tweets, a training set has been defined $D = \{x_1, x_2, \ldots\ldots, x_n\}$ where each record is labeled to a class; the classification model is related to the characteristics in the record underlying one of the class labels; then, for a given instance of unknown class, the model is used to predict a class tag [3].

2.1.1. Supervised

The methods of supervised learning depend on training documents that contain labels, usually these labels are performed by a supervisor manually; This means that the documents or tweets are cataloged or labeled according to the supervisor. Subsequently, these training data are used for the prediction of the class [4].

Among the techniques used for the SA, we have Support Vector Machines, Naïve Bayes, Decision Tree, Neural Networks, and K Nearest Neighbors, among others.

2.1.2. Unsupervised

In supervised learning, the goal is to learn a mapping from the input to an output whose correct values are provided by a supervisor. In unsupervised learning, no such supervisor is presented and we only have input data. The objective is to find the regularities in the entrance [5].

3 Proposal

The aim of this work is to compare learning approaches for the SA in the social network Twitter. For this we select the dataset, preprocessing of text, classification of tweets (Positive or Negative Polarity) will be carried out and finally it will be evaluated every technique to know its performance.

For this work we have several setups in the selection of dataset as well as the preprocessing of tex. Finally, we will perform the learning and evaluation of each method used. In the Figure 1 we show the process we have used. In the next section, more details will be given regarding the tests performed.

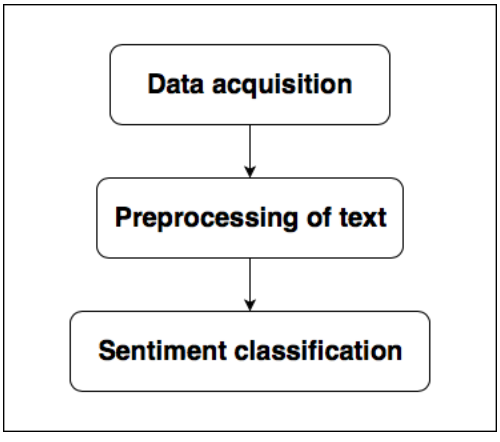


Figure 1: Sentiment analysis process

4 Experimental Setup

4.1 Data acquisition

For the tests we used a manually classified dataset related to the president of Ecuador. The first case, we used 1718 tweets of which 859 are positive and 859 are negative; within the total it has been established that 70% of the data is for training (1201 tweets of which 601 are positive and 600 are negative) and the remaining 30% are for testing (516 tweets, of which 258

are positive and 258 are negative, these data were extracted in the year 2017.

The second case, we used 1718 tweets as training dataset, of which 859 are positive and 859 are negative, these tweets were extracted in 2017 before the election for President of the Republic of Ecuador. Finally, we used a test data, with a total of 100 tweets, of which 50 are positive and 50 are negative, these data were extracted in the current year 2018.

4.2 Preprocessing of text

In this step we implemented several techniques for cleaning and eliminating noise, within the content of each tweet analyzed [6]. Below are listed each:

- Remove WWW within the tweet.
- Remove stopwords in the tweet.
- Remove htttp within the tweet.
- Remove the hashtags.
- Remove punctuation marks.
- Remove multiple blank spaces.

4.3 Learning and Evaluation

To evaluate tweet classification techniques, we used the metric F_1 which has been defined in the following way, for each of the machine learning methods such as: Naive Bayes (*NB*), Neural Network (*NN*), Support Vector Machine (*SVM*), K-Nearest Neighbors (*KNN*) and Decision Tree (*DT*) [7].

$$\mathrm{F1} = \frac{2TP}{2TP + FP + FN}$$

For the evaluation tests, we have considered 70% of the dataset as a training dataset and 30% of the dataset as a test dataset. This distribution is considered since the amount of data is limited considering the analysis that is being made.

5 Experimental Results and Analysis

The tweets used to carry out the tests were extracted in the year 2017 during the electoral campaign to the Presidency of Ecuador, specifically the tweets focus on the candidate Lenin Moreno, which is from the political party Alianza País.

We present below Table 1, which presents the performance of F_1 for each of the 5 machine learning techniques used in this paper. The values are shown ordered from highest to lowest considering the evaluation of each of the techniques.

Methods	F1	F1 without hashtags
NB	83.21%	83.21%
NN	82.68%	82.51%
SVM	82.09%	82.09%
KNN	75.16%	75.16%
DT	68.15%	65.84%

Table 1: Performance measure F_1 for each method – Test #1

The results obtained in Table 1 indicate that Naïve Bayes has the highest score in F_1, when compared to the other techniques, with a value of 83.21%, and the technique with the lowest performance is DT with a value of 68.15% with hashtags, and 65.84% without hashtags.

Methods	F1 with hashtags	F1 without hashtags
NB	78.00%	78.00%
NN	72.34%	70.97%
DT	71.84%	71.43%
SVM	68.97%	68.97%
KNN	44.44%	44.44%

Table 2: Performance measure F_1 for each method - Test #2

For the next test, we used the dataset of 2017 as training set and the dataset of 2018 as a test set. In Table 2 the performance of each of the techniques is shown, in addition to performing a test without applying the hashtags removal. The results obtained in Table 2 indicate that the *NB* technique has the highest score of F_1 when compared to the other techniques, with a value of 78%, and the technique with the lowest performance is *KNN* with a value of 44.44% with both hashtags and without hashtags.

6 Conclusions and Future Work

The sentiment analysis allows, through the processing of natural language, to have an approximate vision of what people are thinking at a given moment. The results obtained from the SA indicate high performance, probably due to the use of a manually classified data set that has several notable characteristics, such as: the period of the tweets, the location of the tweets, and the topic.

Our future work will include the separation of tweets in positive, negative or neutral tweets with a very low polarity. Also, we will use bigrams and trigrams instead of just unigrams. Additionally, we want to compare machine learning techniques focused on other areas such as: marketing, education, culture, among others. In this way we will evaluate the performance and behavior of each technique according to the study domain, since the information generated by Twitter is very varied, making it necessary to analyze several topics or social problems.

Acknowledgments

This work was possible thanks to Senescyt of Ecuador for the financing of research studies at the Polytechnic Institute of Leiria, Portugal.

References

[1] M. VOHRA and J. TERAIYA, “A Comparative Study of Sentiment Analysis Techniques,” Ejournal.Aessangli.in, vol. 17, no. 4, pp. 313–317, 2013.

[2] V. A. Kharde and S. S. Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques,” Int. J. Comput. Appl., vol. 139, no. 11, pp. 975–8887, 2016.

[3] C. H. B. S. Silva and M. R. Bernardete, “Inductive Inference for large scale text classification,” Universidade de Coimbra, 2008.

[4] A. K. Behera, “Performance Analysis of Supervised Machine Learning Techniques for Sentiment Analysis,” pp. 128–133, 2017.

[5] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” Ain Shams Eng. J., vol. 5, no. 4, pp. 1093–1113, 2014.

[6] G. Angiani et al., “A comparison between preprocessing techniques for sentiment analysis in Twitter,” CEUR Workshop Proc., vol. 1748, no. Ml, 2016.

[7] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” Inf. Process. Manag., vol. 45, no. 4, pp. 427–437, 2009.

Paper:#41

Granularity and time window on forecasting regression problems

By: Silva, C. Grilo, C. Silva, C.

Granularity and time window on forecasting regression problems

Cláudio Silva[1]
2141551@my.ipleiria.pt
Carlos Grilo[1,2]
carlos.grilo@ipleiria.pt
Catarina Silva[1,3]
catarina@ipleiria.pt

[1] School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2] CIIC, Polytechnic Institute of Leiria, Portugal
[3] Center for Informatics and Systems of the University of Coimbra, Portugal

Abstract

Load forecasting has different approaches and applications, with the general goal of predicting future load in a period of time ahead on a given system. In this paper the effect of granularity and time window is analysed on two different forecasting regressions problems: web server load prediction and energy load prediction. The tests are conducted with three different learning methods, results on both datasets show the prominent performance on the resulting models.

1. Introduction

Forecasting gives researchers the ability to anticipate events, to better adapt to changes, and to possibly alter outcomes. Load forecasting can be defined as the science or art of predicting the future load in a specific period ahead on a given system [1]. Forecasting applicability is found in many areas, such as, commercial, industrial, scientific, and economic activities [2].

In this work, our focus relies on load forecasting in servers, commonly referred to as server load prediction and also on energy load forecasting.

On both problems the predictions horizons are on the hourly scale, these types of forecasting are commonly referred as short-term load forecasting (STLF) [3].

The two most commonly used approaches for server load prediction and energy load forecasting, are a time series-based approach and a regression approach. In the case of the time series approach, the idea is to predict the load using mainly data with a time dimension. Predictions tend to be daily, weekly or seasonal. Auto-regressive moving average (ARMA) models, Kalman filtering, spectral expansion techniques and the Box-Jenkins methods are common methodologies used in time series approaches [4]. For the regression approach, common methodologies are linear regression, stepwise regression or Lasso regression. Most regression methods try to define the basic functional elements, identify the coefficients needed in the linear combination of the functional elements previously defined and select all the relevant variables [5].

The rest of the paper is organized as follows. Section 2 presents the approach used in this work, in Section 3 the methods along with their configuration are described, followed by Section 4 were the results are presented, the paper is concluded in Section 5.

2. Proposed approach

In this work two different load forecasting problems were used to compare the effects of granularity and time window effects on said regression problems. The first problem used data related to the Wikipedia server activity and had a model created for hourly prediction of the load generated by English written pages only. The second problem used data related to the hourly electricity consumption in the city of Leiria, Portugal, with a model focused also in hourly prediction.

Figure 1 presents the solution architecture applied in this work. The initial step involves creating different datasets with different combinations of granularity, time windows and normalized and non-normalized data. From the newly created datasets a sub-set of training and test set is selected, this sub-set is selected by taking in consideration the first results achieved and their representation of real life situation, e.g., a training and test set with a granularity of one and a time window of 24, this combination when transpose to the real life represent the hourly load with a history of load of 1 day.

The dataset transformation module, involves a set of transformations in the initial dataset. Figure 2 depicts the different processes involved in the transformation. In the first step, the granularity of the data is selected. In this work the granularity is always in an hourly scale. After that, the time window definition is applied. In this step, the number of hours back in time a record has is defined. Since the granularity step was already applied, the created history has the same scale. The last step consists in applying normalization on the resulting dataset. For each created dataset that is normalized, there is another one that remains non-normalized. This is done in order to evaluate how normalization affects the results.

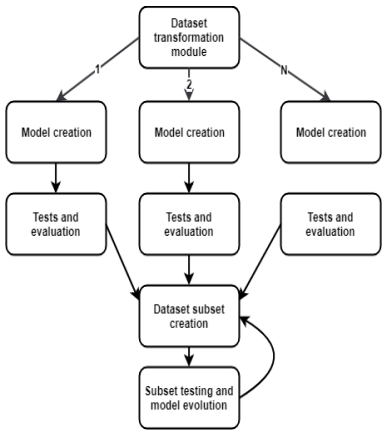


Figure 1: Solution architecture diagram. The dataset transformation module creates different datasets with different time windows and granularity configurations (1 to N)

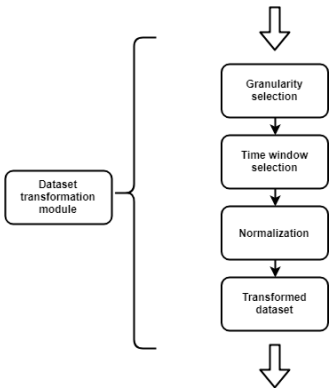


Figure 2: Diagram depicting the different steps involved in the data transformation module

3. Learning methods and experimental setup

The learning methods used in our approach were Linear Regression, Artificial Neural Networks (ANN) and Support Vector Machines (SVM). The use of these specific methods resulted from an extensive literature review. Although other methods can be applied in load forecasting, our selection of learning methods tries to compare a simple learning method (linear regression), which will work as baseline, along with two more complex learning methods (ANN and SVM), already used and tested by a large community of researchers and with already proven performance in the area.

Our linear regression model used the Akaike criterion with attribute selection method M5 and ridge parameter with value 1.0E-8.

The ANN model, used a Multilayer Perceptron architecture with Backpropagation to classify the instances. The configuration used is as follows: learning rate: 0.3; validation threshold of 20; training time of 500 iterations; number of hidden layers defined by the formula hidden layers = (number of input attributes + number of output neurons) / 2 and momentum value: 0.2.

The SVM model was implemented based on the SVM variation SMOreg. The created model used a c value of 1, a Poly Kernel variation with a cache size of 250007 and an e value of 1 and the regression optimizer: RegSMOImproved, with an e of 1.0E-12 and tolerance of 0.001.

All models were trained and evolved focusing in a better Mean Absolute Percentage Error (MAPE). This metric was used in order to have scale independent values for comparison between server load prediction and energy load forecast. A lower MAPE value indicates a smaller difference between the real value and the one predicted by the model. However, the MAPE metric must be used with caution when in pursue of a smaller value, since this could easily lead to an overfitted model.

4. Experimental results

The created models were based in the above-mentioned learning methods, Linear Regression, ANN and SVM, and were tested on normalized and non-normalized datasets.

Our dataset for the energy problem, originated from raw data provided by EDP (Energias de Portugal), it is composed by the electric current attribute and a temporal dimension added to each record, by adding the electric current value of the next (n)-hours that would follow. The dataset used in server load prediction was based on Wikipedia traffic, it is publicly available through the Wikimedia project. The initial data used, was retrieved from the Wikimedia foundation directories. This data contained the collected data/traffic generated by Wikipedia servers. From this data a dataset was created containing the number of request made to the server in an hour and the total generated load resulting from these requests. The number of requests and load were gathered from English only pages.

In the energy load forecasting problem, the data used concerned April 2017. In the server load prediction problem, the used data was from January 2016. For both problems, time windows of 4, 6, 8, 24 and 168 hours and granularity values of 1, 2 and 7 were used, these values were chosen, since they best represent and divide different temporal spectrums in a day or week(s).

Although tests were conducted in normalized and non-normalized data, the results presented are all for normalized data, since the subset creation process led to the conclusion that normalization had no impact in the MAPE value.

Table 1: Results summary on the energy load forecasting problem

Linear Regression

Granularity	Time window				
	4	6	8	24	168
1	9.12	8.89	8.87	8.55	8.57
2	6.67	6.54	6.59	6.68	6.61
7	6.16	6.15	6.16	4.81	9.08

Artificial Neural Networks

Granularity	Time window				
	4	6	8	24	168
1	14.34	9.03	10.71	9.02	13.91
2	8.96	8.02	6.73	8.54	12.08
7	6.98	6.39	5.53	5.25	28.26

Support Vector Machines

Granularity	Time window				
	4	6	8	24	168
1	9.15	9.01	8.96	8.79	8.77
2	6.70	6.57	6.65	6.27	7.16
7	6.25	6.17	6.25	4.82	13.39

Table 1 presents the results for the energy load forecasting problem. The tests show that increasing the time window, decreased the MAPE significantly, however this was only true until the 24-hours mark, after that the MAPE increased, achieving its highest error in our tests with time window of 168 and granularity of 7.

When applying a granularity value to the energy problem, higher granularity values revealed to achieve lower MAPE values. For all 3 learning methods it was possible to verify a trend of lower MAPE values for higher granularity, although with some outliers.

The best result, 4.81%, was found with the 24-hour time window and a granularity of 7, using linear regression.

With granularity of only 2 the average error was 7.414%. When using granularity of 1 and 7 the average error was 9.756 % and 9.823% respectively. These average results however are deceiving, since the standard deviation with granularity 7 is extremely high, as a common rule it is possible to affirm that increasing granularity is justifiable but just with time window of 24 or lower.

Table 2: Results summary on the server load prediction problem

Linear Regression

Granularity	Time window				
	4	6	8	24	168
1	1.191	1.18	1.176	1.143	1.458
2	1.311	1.279	1.247	1.186	3.198
7	1.259	1.336	1.211	1.201	3.807

Artificial Neural Networks

Granularity	Time window				
	4	6	8	24	168
1	1.157	1.269	1.387	1.236	2.039
2	1.353	1.461	1.285	1.283	3.12
7	1.834	1.655	1.989	2.284	9.753

Support Vector Machines

Granularity	Time window				
	4	6	8	24	168
1	1.187	1.183	1.18	1.114	1.271
2	1.387	1.369	1.276	1.253	2.628
7	1.392	1.281	1.244	1.238	3.801

Table 2 presents the results for the server load prediction problem. It can be seen that increasing time window decreases the MAPE value, but just until the 24-hour mark. After that, our research revealed that the MAPE values increased significantly, as confirmed by the 168- hour mark. This pattern was found in all three learning methods.

As for granularity, although its impact in the MAPE value when compared to the time window is less representative, it is possible to verify an increase in error when its value is not equal to one. Different from time window, a trend with increase in granularity is not possible to verify.

The smallest MAPE value (1.114%) was found with time window of 24 hours, granularity of 1 and using SVM. The experiments revealed that, on average, Linear Regression and SVMs presented better and nearly equivalent results, while NN results are not as competitive.

5. Conclusion

This paper describes an approach to the load forecasting problem using granularity and time window as its core basis. The use of two different load forecasting problems enabled the comparison of the applicability of granularity and time window. In the two load forecasting problems, equivalent results were achieved when using the time window value, which in, both cases, gradually decreased the MAPE until the 24-hour mark. However, in terms of granularity a similar trend in both problems was not possible to verify. In the server load prediction problem, the best results were achieved with granularity 1 while in the energy load forecasting problem the results performance increased with the increase of the granularity value.

For the future, improvements in our study could be made in terms of data from more months, as well as source. That is, data from different problems would potentially help disclosing a more generic conclusion in terms of the influence of granularity and time window on regression problems.

References

[1] S. A. hady Soliman, Electrical load forecasting: modelling and model construction. Oxford: Elsevier Inc., 2010.

[2] [2] C. Chatfield, TIME-SERIES FORECASTING. London: Chapman and Hall, 2000.

[3] Gross, George, and Francisco D. Galiana. "Short-term load forecasting." Proceedings of the IEEE 75.12 (1987): 1558-1573.

[4] Park, D.C., M.A. El-Sharkawi, R.J. Marks, L.E. Atlas and M.J. Damborg, “Electric Load Forecasting Using an Artificial Neural Network”, IEEE Transactions on Power Engineering, Seattle, 1991.

[5] Pole, Andy, Mike West, and Jeff Harrison. Applied Bayesian forecasting and time series analysis. Chapman and Hall/CRC, 1994.

Paper:#43

Twitter message: is bigger the better for classification purposes?

By: Costa, J. Silva, C. Ribeiro, B.

Twitter message: is bigger the better for classification purposes?

Joana Costa[12]
joana.costa@ipleria.pt,joanamc@dei.uc.pt
Catarina Silva[12]
catarina@ipleiria.pt,catarina@dei.uc.pt
Bernardete Ribeiro[2]
bribeiro@dei.uc.pt

[1] School of Technology and Management
Polytechnic Institute of Leiria, Portugal
[2] CISUC - Department of Informatics Engineering
University of Coimbra, Portugal

Abstract

Last year Twitter doubled the size of the tweet. Character count initially set to 140 character was then expanded to 280 characters and, according to the Twitter official blog, the idea was reducing the effort of fitting the message into the tweet, and thus minimizing the problem of abandoning tweets before sending, due to the time spent in editing. However, and considering the relevance of information extraction from such media, it is important to understand the impact of such change in Twitter message classification. In this work, we present the effect of doubling the size of a tweet in the scope of a Twitter classification problem. Results on a real dataset suggest that the bigger tweets are, the harder they are to classify, making the scenario more challenging.

1 Introduction

Social networks are becoming a major hub for spreading and gathering information[1,2,5]. Machine learning pattern recognition techniques are now pervasive in such tasks. Nevertheless, algorithms and models are created amidst constant changes in the underlying system, whether it may be privacy rules or available features, or, as we will delve into in this work the number of characters available for a post.

When such changes occur, previous models usually either completely fail to deliver the desired service or, at least, experience a noticeable degradation. In this work, we present a study on how current models behave and how they could react to this paramount change in Twitter from 140 to 280 characters.

The rest of the paper is organized as follows. Section 2 introduces background on Twitter and Section 3 formulates the classification problem. Section 3 describes the proposed approach and Section 4 details the experimental setup. Section 5 presents the experimental results and analysis and, finally, Section 6 concludes and delineates future research lines.

2 Twitter

Twitter, created in 2006, rapidly gained popularity. According to the company, its mission is to give everyone the power to create and share ideas and information instantly, without barriers. Nevertheless, the concept can be easily described as a network where a user can share a simple message, which is immediately made public.

Twitter took advantage of the worldwide implemented Short Message Service (SMS), and promoted the idea of sharing simple day life events in order to stay connected with friends and family.

One of the most significant differences of Twitter as a social network is that the relation between users in not necessarily reciprocal, i.e., in Twitter a user can follow another user without being followed back. Another distinctive characteristic of Twitter is that, by default, all messages are public.

2.1 Tweet

A *tweet* is any message posted to Twitter which may contain photos, videos, links and up to 140 characters of text. There is also the concept *retweet*, described as a tweet that you forward to your followers. Tweets can also include *mentions*, a user name started with the symbol "@", and *hashtags*, detailed in the following. An example of a tweet is shown in Figure 1.

Figure 1: Example of a tweet

2.2 Hashtag

Twitter provides the possibility of including a *hashtag*. A hashtag is a single word starting with the symbol "#", as represented in Figure 2. It is used to classify the content of a message and improve search capabilities. This can be particularly important considering the amount of data produced in the Twitter social network. Besides improving search capabilities, hashtags have been identified as having multiple and relevant potentialities, like promoting an idea, behavior, or style, that spreads from person to person within a culture. By tagging a message with a trending topic hashtag, a user expands the audience of the message, compelling more users to express their feelings about the subject[7].

Figure 2: Tweet and hashtag representation

3 Twitter classification problem

The classification of Twitter messages can be described as a multi-class problem. Twitter messages, represented as $\mathcal{D} = \{d_1, \ldots, d_t\}$, where d_1 is the first instance and d_t the latest. Each instance is characterized by a set of features, usually words, $\mathcal{W} = \{w_1, w_2, \ldots, w_{|\mathcal{W}|}\}$. Consequently, the instance d_i is represented by the feature vector $\{w_{i1}, w_{i2}, \ldots, w_{i|\mathcal{W}|}\}$.

If d_i is a labelled instance it can be represented by the pair (d_i, y_i), where $y_i \in \mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}|}\}$ is the class label for instance d_i.

Our classification strategy, presented in [2], will be using the Twitter message hashtag to label the content of the message, which means that y_i represents the hashtag that labels the Twitter message d_i.

Notwithstanding being a multi-class problem in its essence, it can be decomposed in multiple binary tasks in a one-against-all binary classification strategy.

4 Proposed approach

The key factor in our approach is to determine if tweets that benefited from the expansion to 280 characters are easier to classify. We propose two different learning models, one that uses only tweets with less than 140 characters, and another that used tweets bigger than 140 characters.

Figure 3 presents the proposed approach that is divided then into two parts:

1. Learning Model 140: both train and test sets include only tweets up to 140 characters
2. Learning Model 280: both train and test sets include only tweets between 140 and 280 characters

This proposal allows for an independent comparison of learning tweets' classification with different lengths.

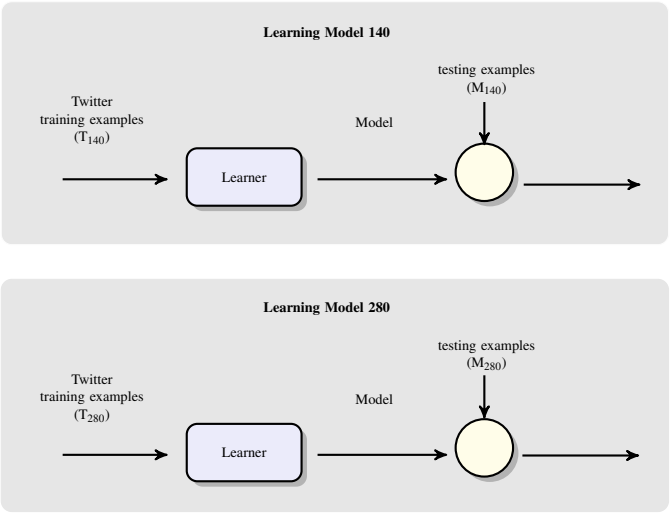


Figure 3: Proposed Approach

5 Experimental setup

5.1 Dataset

To evaluate and validate our strategy, and considering the lack of a labelled dataset with the needed characteristics, we have built a specific dataset. The dataset was constructed by requesting public tweets to the Twitter API . We have collected more than 1,055,000 messages, since 10 March 2018 to 5 April 2018, and, considering the worldwide usage of Twitter, tweets were only considered if the user language was defined as English. All the messages that did not have at least one hashtag were discarded, as the hashtags are assumed as the message classification. Finally, tweets containing no message content besides hashtags were also discarded and all the hashtags are removed from remaining tweets. From the 1,055,000 collected messages, we reach 142,000 tweets that have a body part and at least one hashtag.

We have only considered five of the most popular used hashtags, namely #syria, #sex, #trump, #oscars and #ucl.

The tweets were then split into two equal and disjoint sets: training and test. The data from the training set is used to select learning models, and the data from the testing set to evaluate performance. We have used the bag of words strategy to document representation along with tf-idf. Preprocessing methods were applied, namely stopword removal and stemming.

5.2 Learning models

The validate our approach, we have used the previously described dataset and Support Vector Machine (SVM)[6], based on the Statistical Learning Theory and Structural Risk Minimization Principle. The idea behind the use of SVM for classification consists on finding the optimal separating hyperplane between the positive and negative examples. Once this hyperplane is found, new examples can be classified simply by determining which side of the hyperplane they are on. SVM constitute currently the best of breed kernel-based technique, exhibiting state-of-the-art performance in text classification problems[3]. A linear kernel was used.

5.3 Evaluation

In order to evaluate the binary decision task of the proposed models we defined well-known measures based on the possible outcomes of the classification, such as, error rate ($\frac{FP+FN}{TP+FP+TN+FN}$), recall ($R = \frac{TP}{TP+FN}$), and precision ($P = \frac{TP}{TP+FP}$), as well as combined measures, such as, the van Rijsbergen F_β measure, which combines recall and precision in a single score: $F_\beta = \frac{(\beta^2+1)P \times R}{\beta^2 P+R}$. F_β is mostly used in text classification problems with $\beta = 1$, i.e. F_1, an harmonic average between precision and recall. Micro-averaged F_1 is computed by summing all values (TP, FP, TN and FN), and then use the sum of these values to compute a single micro-averaged performance score that represents the global score.

6 Experimental results and analysis

We evaluate the performance obtained on the Twitter data set using the two approaches described in Section 4, namely the Learning Model 140 and the Learning Model 280. Figure 4 represents graphically the performance results obtained by classifying the dataset, considering the micro-averaged F_1 measure.

Analysing the graph we can observe that globally, and considering the average of the micro-averaged F_1, longer tweets are more difficult to classify. In all represented classes, longer tweets are classified with a worst classification performance than shorten tweets. These might seem strange as more words would turn the tweet more informative, but that diversity might be counter-productive in terms of classification, as tweets might not be so straight to the point considering its content. In some classes, like those represented by the hashtags #sex and #ucl, the difference it not so significant (less than 1% F_1), but in classes like the represented with hashtag #oscars the difference is almost 7%. It is also important to note that the difference in the classification performance is bigger in classes more difficult to classify, as those with a classification performance higher tend to have slight differences in the classification performance of tweets with less than 140 characters and more than 140 characters.

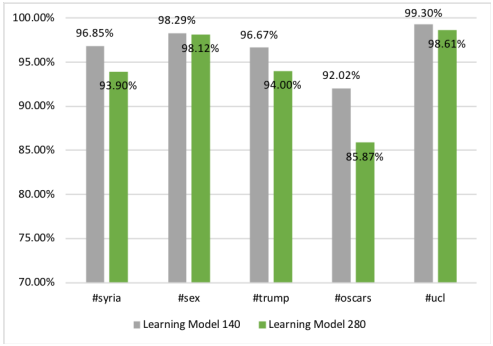


Figure 4: Micro-averaged F_1

7 Conclusions and future work

We have proposed a method to determine the impact of longer tweets in the classification performance of Twitter classification messages. Shorten tweets deemed better than longer tweets and are easier to classify. Although in some classes the difference is not significant, others, specially those harder to classify, the difference can be almost 7%, which might request that those tweets might need a specially approach in classification problems.

Our future work will include a more profound study about the characteristics of longer tweets that turn them harder to classify, specially perceiving if it is possible to mine information that bag of words is not capable of, like taking use of semantics.

References

[1] Eli Bartov, Lucile Faurel, and Partha S Mohanram. Can twitter help predict firm-level earnings and stock returns? The Accounting Review, 93(3):25-57, 2017.

[2] Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro. Defining semantic meta-hashtags for Twitter classification. In Proc.11th Int. Conference on Adaptive and Natural Computing Algorithms, pages 226-235, 2013.

[3] T. Joachims. Learning Text Classifiers with Support Vector Machines. 2002.

[4] Atul Nakhasi, Sarah G Bell, Ralph J Passarella, Michael J Paul, Mark-Dredze, and Peter J Pronovost. The potential of twitter as a datasource for patient safety. Journal of patient safety, 2018.

[5] Sara Rosenthal, Noura Farra, and Preslav Nakov. Semeval-2017 task4: Sentiment analysis in twitter. In Proceedings of the 11th Inter-national Workshop on Semantic Evaluation (SemEval-2017), pages 502-518, 2017.

[6] V. Vapnik. The Nature of Statistical Learning Theory. 1999.

[7] Michele Zappavigna. Ambient affiliation: A linguistic perspective on Twitter. New Media Society, 13(5):788-806, 2011.

Paper:#44

Automatic music transcription using a one-classifier-per-note approach

By: Gil, A. Reis, G. Domingues, P. Grilo, C.

Automatic music transcription using a one-classifier-per-note approach

André Gil[1]
2151630@my.ipleiria.pt

Gustavo Reis[1,2]
gustavo.reis@ipleiria.pt

Patrício Domingues[1,2,3]
patricio.domingues@ipleiria.pt

Carlos Grilo[1,2]
carlos.grilo@ipleiria.pt

[1]School of Technology and Management, Polytechnic Institute of Leiria, Portugal

[2]CIIC, Polytechnic Institute of Leiria, Portugal

[3]Instituto de Telecomunicações, Portugal

Abstract

This paper describes a new approach to the automatic music transcription problem. The architecture of this approach consists on an artificial neural network per each possible note, plus an additional one (per note) for post-processing. We refer to the main artificial neural networks as *classifiers* and the additional ones, used for post-processing, as *post-processing units*. From our knowledge, it is the first time that a comparison of several classifiers with the traditional one-single classifier approach is done. In addition, to the best of our knowledge, it is the first time that an artificial neural network is applied as a post-processing method in the problem of automatic music transcription.

Keywords: Automatic Music Transcription, Multi-Pitch Estimation, Artificial Neural Networks

1 Introduction

Automatic music transcription (AMT) is the process of detecting the notes that are present in a musical piece, via a machine. This work focuses on a variant of this problem, called multi-pitch estimation, which consists in identifying the pitched notes present in a polyphonic musical piece. A common method applied to this problem is *artificial neural networks* (ANNs). ANNs have been applied successfully to several complex problems. In this paper we present a novel approach of automatically transcribing piano music by using several ANNs.

The traditional approach to AMT, especially when ANNs are applied, consists in having a single classifier that detects and transcribes all the musical notes of a piano (Figure 1a) [1], [2] and [3]. However, in this work, several classifiers are created, each one responsible for detecting and transcribing a specific musical note (Figure 1b). The rationale behind this idea is the well-known "divide and conquer" approach, where a larger hard problem is divided into smaller sub-problems. Hypothetically, these sub-problems should be easier to solve than the original one, possibly boosting the performance of the whole AMT system.

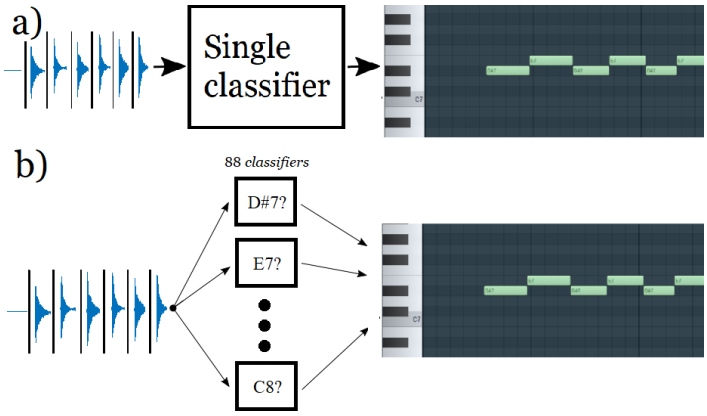


Figure 1 - Illustration of a) the traditional single all-notes classifier approach and b) the many one-note classifiers approach.

In general, the AMT process starts by splitting the musical piece wave (time domain), into smaller chunks, called frames. For each frame, 4096 samples are taken so that a frequency domain signal is produced. Given that the second half of this signal mirrors the first half, only the first 2048 values are taken as input for the notes classifier. The classifier (in our case, a set of classifiers) gives as output, the musical notes that are believed to be present in the frame.

After the classification process, errors are common. This means that some notes that are not present in the frame are identified as being there and/or some notes that are in the frame are not identified. Post-processing techniques can be used to correct these mistakes. We also propose an additional ANN based post-processing step.

The rest of the paper is organized as follows: Section 2 presents the proposed model, while Section 3 presents and discusses the main results. To finish, conclusions and future work are given in Section 4.

2 Proposed model

In this section a deeper analysis of the proposed model is presented. This analysis is composed by four major parts: dataset creation, pre-processing, classifiers and post-processing.

2.1 Dataset

To be able to compare our approach to already existent ones, we use the same musical pieces from *Configuration 1*[1], based on the MAPS [4] dataset. *Configuration 1* consists in four folds, each one with 216 musical pieces for training and 54 pieces for testing. As a result, since AMT usually works with 88 musical notes (corresponding to the notes of a grand piano), we have 88 classifiers and 88 post-processing units per fold. This results in a total of 352 classifiers and 352 post-processing units. Accordingly, each classifier will also have its own specific dataset.

2.2 Pre-processing

Regarding ANNs, a fundamental key point to consider is that good quality data is needed to achieve good results. For this purpose, three sequential transformations in the data have been done: (1) transform the input data into the time-frequency domain; (2) removal of meaningless data, like frames with silence; (3) choosing the best ratio between frames with and without a specific musical note.

2.3 Classifiers

The classifiers consist on the traditional feed-forward neural networks. Other more recent techniques could have been chosen as, for example, Convolutional Neural Networks. However, we wanted to have a baseline to be used in the future when experimenting with more powerful techniques.

After preliminary experiments, where a vast variety of hyperparameters and optimization techniques combinations were tested, the chosen classifiers configuration was the following: The ANN has 2048 inputs (samples from the signal FFT), 5 hidden layers, where each layer has the size of *previous_layer_size/2*, with the first one having 256 units and an output layer with one unit (yes or no output). Hidden layers units apply the *leaky relu* activation function and the output layer uses the *sigmoid* function. The optimizer chosen is *Adam* [5] and the *learning rate* is $1e^{-6}$. Also, the loss function used is the *cross entropy*. Finally, the optimization techniques used are the *dropout* [6] with a probability of 15%, noisy gradients [7] with probability of 70% and a standard deviation of 0.05. Data is also shuffled [8] each iteration (music pieces sequence and frames shuffling).

In the end, these hyperparameters and optimization techniques improved significantly the performance of the classifiers not only in terms of evaluation metrics but also in reducing the needed time for training.

[1] More details at: http://www.eecs.qmul.ac.uk/~sss31/TASLP/info.html.

2.4 Post-processing

As mentioned earlier, a post-processing method can correct errors from the classifier or set of classifiers, which in turn impacts significantly the results of an AMT system. In this work we applied an ANN for this task. The motivation is due to two main factors: firstly, ANNs are good problem solvers and secondly, to our best knowledge, they have never been applied as a post-processing method in the AMT problem.

These post-processing units were trained with the outputs of a classifier. Since the output layer of each classifier applies the sigmoid function as activation function, the output consists in a probability value. As a result, for each frame, a probability value is given. From these outputs, a sequence of 9 elements is then used in order to determine the value in the middle of that sequence (9 inputs and 1 output). The result is an ANN that, not only considers the current frame, but also the previous and following four frames, in order to verify whether the same musical note is in the middle frame or not.

We created three types of post-processing units: (1) *one for all*, where a single unit would be responsible to post process all the notes, (2) *one per each*, where a post-processing unit would be responsible to post process a specific note, (3) *improved one per each*, which is similar to (2) but where the input data is previously transformed with several pre-processing techniques. These pre-processing techniques consist in shuffling the sequence of frames, as well as, an adequate balance between middle frames that contain a positive correct label, frames with a negative correct label and wrongly classified frames.

Post-processing units are trained once the classifiers have been trained. The motivation for this is that we wanted a stable output result from the classifiers, so that post-processing can adapt quickly to each classifier. Each post-processing unit was trained using the same hyperparameters and optimization techniques as the classifiers, except for the learning rate, that had a value of $1e^{-5}$, and the model architecture of the ANN. Three hidden layers were used, each one with five neurons.

From our preliminary experiments, we were able to conclude that the post-processing unit *one for all* was not able to successfully improve the results of the whole system. On the other side, both post-processing units, *one per each* and *improved one per each* significantly improve the global performance of the system. Nevertheless, the best one was the *improved one per each*, especially with musical notes that are not so frequent in the dataset. Therefore, this filter is being used as our post-processing method in our final experiments.

3 Results

In this section, results and a comparison with similar techniques used in other state of the art works are presented. Initially, the metrics used for comparison are introduced. Then, the results from our model per each fold are shown, and finally, a comparison with other research works is presented.

3.1 Metrics

To compare our model, we have taken into consideration frame-based metrics [9], more specifically, *precision* (P), *recall* (R) and *f-measure* (F). These evaluation metrics consist in comparing the transcribed binary output with the MIDI ground truth, frame by frame. Mathematically this can be expressed as:

$$P = \sum_{t=1}^{T} \frac{TP[t]}{TP[t] + FP[t]} \quad (3.1)$$

$$R = \sum_{t=1}^{T} \frac{TP[t]}{TP[t] + FN[t]} \quad (3.2)$$

$$F = \frac{2 * P * R}{P + R} \quad (3.3)$$

where *TP[t]* represents the number of true positives for the event (frame) at *t*, *FP* is the number of false positives and *FN* is the number of false negatives.

3.2 Comparison with other approaches

We compare our results with the ones achieved in [1] and [2], two state of the art works. Both works used *Configuration 1* as their dataset. The results are presented in Table 1. We only present results obtained with the classifiers, because final results with post-processing are not available yet. However, preliminary experiments show that post-processing improves the results.

The results show that our approach has slightly better results regarding recall. Results are not as good for precision and F-measure when compared to the other two works, although they are close to the ones obtained in [1], where post-processing is used. We consider these results as promising, given that they are a first approach, and we expect that they improve after adding post-processing.

Table 1 - Results of the average of the four folds from *Configuration 1*.

Model	***P***	***R***	***F***
Hybrid DNN [1]	65.66	70.34	67.92
Framewise DNN [2]	**76.63**	70.12	**73.11**
One-classifier-per-note ANN	62.08	**72.06**	66.65

4 Conclusions and future work

In this paper, we present a novel *divide and conquer* approach to tackle the AMT problem. Our first results show that this is a promising path, since the results are close to other state of the art works. Indeed, preliminary experiments show that the ANN based post-processing process described in the paper is able to improve the results.

To conclude, there is plenty of space for future work. For instance, in the case of the classifiers, other techniques can be used, like *recurrent neural networks* or *convolutional neural networks*. In addition, the post-processing units could also use other type of technique or, instead, incorporate the transcription results from other classifiers to better extract patterns from the data.

References

[1] S. Sigtia, E. Benetos and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(5), 927-939, 2016.

[2] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt and G. Widmer. On the potential of simple framewise approaches to piano transcription. arXiv preprint arXiv:1612.05153, 2016.

[3] R. Kelz and G. Widmer. An experimental analysis of the entanglement problem in neural-network-based music transcription systems. arXiv preprint arXiv:1702.00025, 2017.

[4] V. Emiya, R. Badeau and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. IEEE Transactions on Audio, Speech, and Language Processing, 18(6): 1643-1654, 2010.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1): 1929-1958, 2014.

[7] Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. Adding gradient noise improves learning for very deep networks. arXiv preprint arXiv:1511.06807, 2015.

[8] Bottou, Y. L. L., and GO, M. Efficient backprop. In Neural Networks: Tricks of the trade, Springer, 1998.

[9] Bay, M., Ehmann, A. F. and Downie, J. S. Evaluation of Multiple-F0 Estimation and Tracking Systems. In ISMIR, pages 315-320, 2009.

Author Index

http://recpad2018.dei.uc.pt