UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

# End-to-End Photoplethysmography-based Biometric Authentication System by using Deep Neural Networks


**Guillem Cortès Sebastià**

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya

Technical Supervisor: Jordi Luque Serrano, Ph.D.
Telefónica I+D Research

Academic Supervisor: Prof. Antonio Bonafonte, Ph.D.
Universitat Politècnica de Catalunya


In partial fulfilment of the requirements for the degree in
*Telecommunications Technologies and Services Engineering*
Major in *Audiovisuals Systems*


UPC · ETSETB

Barcelona, June 2018

*"Look up at the stars and not down at your feet.*
*Try to make sense of what you see, and wonder about what makes the universe exist.*
*Be curious."*

Stephen Hawking

# Acknowledgements

First of all, I am very thankful to Jordi Luque and Antonio Bonafonte for their useful guidance throughout the whole project. I would like to thank Antonio for making this project possible in the first place, as it was his idea to get in touch with Jordi at Telefónica I+D. I also want to thank Jordi for his commitment to my learning experience, and for doing so in a challenging and easy-going way.

This work would not have been possible without the help of many people at Telefónica I+D. I would like to thank Joan Fabregat, Javier Esteban and Aleixandre Maravilla for creating the PulseID project; Carlos Segura for his always-accurate comments and tips; and the rest of the researchers, interns and staff for building a great work environment and for all the fun moments.

I would also like to dedicate this thesis to all the teachers I've had throughout my life that have encouraged me to give always my best, to be curious about everything and to overcome adversity.

And last but not least, I woud like to thank to my awesome family that have taught me the importance of the perseverance and not giving up even in the tough conditions. Jordi, Marta, Mariona, gràcies per animar-me sempre a tirar endavant; i Mar, gràcies per creure en mi i estar sempre al meu costat.

# Abstract

Whilst research efforts have traditionally focused on Electrocardiographic (ECG) signals and handcrafted features as potential biometric traits, few works have explored systems based on the raw photoplethysmogram (PPG) signal.

This work proposes an end-to-end architecture to offer biometric authentication using PPG biosensors through Convolutional Neural Networks. We provide an evaluation of the performance of our approach in two different databases: Troika and PulseID, the latter a publicly available database specifically collected by the authors for such a purpose.

Our verification approach through convolutional network based models and using raw PPG signals appears to be viable in current monitoring procedures within e-health and fitness environments, and shows a remarkable potential as a biometric identifier. When tested on a verification task with one second trials, the approach achieved an AUC of 78,2 % and 83,2 %, averaged among target subjects, on PulseID and Troika datasets respectively. Our experimental results on other small datasets support the usefulness of PPG-extracted biomarkers as viable traits for multi-biometric or standalone biometrics. Furthermore, the approach results in a low input throughput and complexity that allows for continuous authentication in real-world scenarios and implementation in little wearable devices. Nevertheless, the reported experiments also suggest that further research is necessary to develop a definitive system.

# Resum

Si bé els esforços en la investigació s'han centrat tradicionalment en senyals electrocardiogràfics (ECG) i característiques artesanals com a trets biomètrics potencials, pocs treballs han explorat sistemes basats en el senyal fotopletogràfic (PPG).

Aquest treball proposa una arquitectura d'extrem a extrem per oferir autenticació biomètrica mitjançant biosensors PPG a través de xarxes convolucionals. L'acompliment d'aquest enfocament s'ha avaluat en dues bases de dades diferents: Troika i PulseID, aquesta última disponible públicament i que ha estat recollida pels autors per a aquest propòsit.

Aquest enfocament de verificació a través de models basats en xarxes convolucionals i l'ús de senyals de PPG en cru sembla ser viable en els procediments de monitorització actuals, dins d'entorns de salut i esport, mostrant així un gran potencial i atractiu per a la biometria. L'enfocament provat en la tasca de verificació, en assaigs que duren un segon, aconsegueix una AUC de $78,2\%$ i $83,2\%$ en mitjana, entre els subjectes objectiu, en els conjunts de dades de PulseID i Troika, respectivament. Els nostres resultats experimentals en altres conjunts petits de dades recolzen la utilitat dels biomarcadors extrets de PPG com a trets viables per a la biometria multi-biomètrica o autònoma. A més, l'enfocament permet una autenticació contínua degut a la baixa complexitat i nombre d'operacions, que la fan sostenible pels escenaris del món real així com per a ésser implementat en dispositius de reduit tamany i capacitat computacional. No obstant això, els experiments reportats també suggereixen que més investigacions són necessàries per a poder desenvolupar un sistema definitiu.

# Resumen

Si bien los esfuerzos en la investigación se han focalizado tradicionalmente en las señales electrocardiográficas (ECG) y características extraídas manualmente como rasgos biométricos potenciales, pocas operaciones han explorado sistemas basados en la señal fotopletográfica (PPG).

Este trabajo propone una arquitectura de extremo a extremo para ofrecer autenticación biométrica mediante biosensores PPG a través de redes convolucionales. Ésta aproximación se ha evaluado en dos bases de datos diferentes: Troika y PulseID, ésta última disponible públicamente y que ha sido recogida por los autores para este propósito.

La verificación a través de modelos basados en redes convolucionales y el uso de señales PPG en crudo parecen ser viables en los procedimientos de seguimiento actuales, dentro del entorno de la salud y del deporte, mostrando así un gran potencial para la biometría. El trabajo testeado en la tarea de verifiación, en ensayos de un segundo, consiguen una AUC de $78,2\,\%$ y $83,2\,\%$ en media, entre todos los sujetos objetivo, en los conjuntos de datos PulseID y Troika respectivamente. Los resultados experimentales en otros conjuntos de datos pequeños refuerzan la potencial utilidad de estos biomarcadores extraídos de señales PPG como rasgos viables para la caracterización biométrica. Además, este enfoque permite una autenticación contínua debido a su baja complejidad y número de operaciones, haciéndola sostenible para escenarios del mundo real así como para poder ser implementado en dispositivos de reducido tamaño y capacidad computacional. Sin embargo, los experimentos aquí reportados sugieren que son necesarias más investigaciones para poder desarrollar un sistema definitivo.

# Revision history and approval record

| Revision | Date | Purpose |
|---|---|---|
| 0 | 21/03/2018 | Document creation |
| 1 | 18/04/2018 | Document revision |
| 2 | 12/05/2018 | Document revision |
| 3 | 13/06/2018 | Document revision |
| 4 | 01/07/2018 | Document approval |

DOCUMENT DISTRIBUTION LIST

| Name | e-mail |
|---|---|
| Guillem Cortès | guillem.cortes.sebastia@alu-etsetb.upc.edu |
| Antonio Bonafonte | antonio.bonafonte@upc.edu |
| Jordi Luque | jordi.luqueserrano@telefonica.com |

| Written by: | | Reviewed and approved by: | | | |
|---|---|---|---|---|---|
| **Date** | 02/07/2018 | **Date** | 02/07/2018 | **Date** | 02/07/2018 |
| **Name** | Guillem Cortès | **Name** | Antonio Bonafonte | **Name** | Jordi Luque |
| **Position** | Project Author | **Position** | Academic Supervisor | **Position** | Technical Supervisor |

# Contents

# List of Figures

# List of Tables

# Nomenclature

| | |
|---|---|
| ADC | Analog to Digital Converter |
| AI | Artificial Intelligence |
| AUC | Area Under the Curve |
| CNN | Convolutional Neural Network |
| DL | Deep Learning |
| DNN | Deep Neural Network |
| ECG | Electrocardiography |
| EEG | Electroencephalography |
| FMR | False Match Rate |
| FNMR | False Non-Match Rate |
| LED | Light Emitting Diode |
| MAC | Multiplier-Accumulator |
| ML | Machine Learning |
| NN | Neural Network |
| PPG | Photoplethysmography |
| PR | Precision-Recall |
| ReLU | Rectified Linear Unit |
| RL | Relational Learning |
| ROC | Receiver Operating Characteristic |
| SGD | Stochastic Gradient Descent |

# Chapter 1

# Introduction

## 1.1. Statement of purpose

With the evolution of the technology, the proliferation of large-scale computer networks, the increasing possibilities that these offers and the concern for identity theft problems, the design of the authentication systems is becoming more and more transcendent. And nowadays, with the facilities we have been used to, it is not enough to design secure systems. The authentication process must be accurate, rapid, reliable, cost-effectively, user-friendly, without invading privacy rights or being too invasive and not supposing drastic changes to the existing infrastructures.

The traditional authentication systems make use of either a secret, personal key (e.g., password, code) and/or a physical token (e.g., ID card, key) that are assumed to be used only by the legitimate users. The problem with the traditional authentication systems is that assumption. It is impossible to ensure hundred percent that the person that is using this password, code, etc. is the genuine person.

However, the biometrics-based personal authentication systems use physiological and/or behavioural traits extracted from the individuals (e.g., fingerprint, iris, voice, face, palmprint, keystroke, mouse, ...) that supposes a way to be sure of the identity of the person who is authenticating. As a result, they are more reliable since biometric information cannot be lost, forgotten, or guessed easily. The authentication accuracy is improved because the biometric traits are stronger than the classic eight-character password in terms of security (Ogbanufe and Kim, 2018). They also improve the user convenience since there is nothing to remember or carry. Nevertheless, the anatomical traits introduced before are exposed to the world and with this, the possibility for a theft to get them and create a copy (T. Fox-Brewster, 2017) (e.g., fake fingerprints,

contact lenses, etc.). Thus, it is important to find new biometric traits which cannot be forged.

Recently, many studies (see Section 2.1) have shown potential of the heart pulse as a biometric trait. It has the advantages of the classic biometric traits and furthermore, it is not exposed. That means that is very hard to get someone's heart pulse.

The goal of this thesis is to develop an End-to-End system able to authenticate persons by using Convolutional Neural Networks (CNN) from their Photoplethysmograph (PPG) signal. This thesis supposes the first approach to an authentication system formed by CNN that verifies the subject identity from it's raw PPG.

This project has been carried out at Telefónica I+D (2018) during the 2017 Fall semester as a contribution to the *PulseID* project.

## 1.2. Requirements and specifications

Since the requirements and the specifications are slightly different, here are presented separately. The main requirement that this project must satisfy is to be able to authenticate person's identity, and do it within this conditions:

- ◇ High-security system. Lowest False Positive Rate (FPR).
- ◇ Good user experience. It has to be usable, so the time required to authenticate is limited.
- ◇ Minimum accuracy of 70%.
- ◇ Authentication algorithm has to be able to run in a Raspberry Pi model. Since this application demands that the algorithm must be implemented in a wearable device, it doesn't have to need huge resources.

Furthermore, the project specifications are the following ones:

- ◇ New dataset of 30 subjects oriented for biometric authentication.
- ◇ Identity verification in 3 seconds.
- ◇ Usage of the Deep Learning.

## 1.3. Methods and procedures

This project aims to present a novel biometric authentication system and the release of the PulseID dataset[1].

The models have been trained and tested using the Troika public dataset (Zhang et al., 2015) and the PulseID dataset developed. Troika dataset contains PPG and ECG signals from several users in a noisy condition as running in a treadmill is. It is worth to mention that this is a dataset created for heart rate tracking and not for biometric authentication. Due to this, it was necessary to create a specific dataset appropite for our research purposes (see Section 3.2).

This project has been developed using PYTHON 3 as the programming language of choice. In addition, we have used several libraries: *NumPy*, *TensorFlow* (Abadi et al., 2015) and *Keras* (Chollet et al., 2015). We have used *Keras* as a deep learning framework using *TensorFlow* as a back-end. Additionally, some BASH scripting has been used in the testing stage. All developed models have been trained on GPU-accelerated servers from Telefónica I+D.

The code developed can be found in a public repository (Cortès et al., 2018) and can be used under Apache 2 license (The Apache Software Foundation, 2018).

## 1.4. Work Plan

The project has been planned into several packages, detailed in Section 1.4.1. The planning in this section corresponds to the latest one, as the initial plan had to be modified. The reasons for this change will be explained in section 1.5.

### 1.4.1. Work Packages

- **WP1** - Documentation
- **WP2** - Prototype
- **WP3** - Dataset
- **WP4** - Software
- **WP5** - Reporting

[1]The PulseID dataset is available upon request from the authors of (Luque et al., 2018) and agreement of EULA for research purposes

These Work Packages (WP) are detailed in the Gantt diagram of section 1.4.2. Each task in each WP is depicted in the diagram.

### 1.4.2. Gantt Diagram

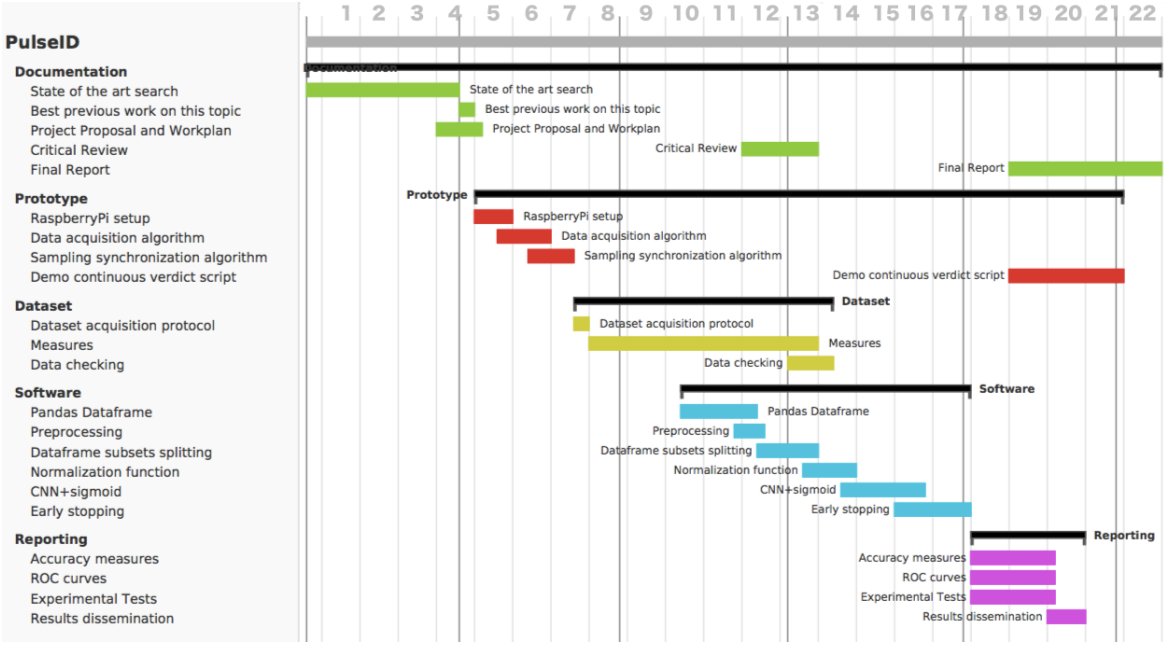



Figure 1.1: Gantt diagram of the project

## 1.5. Incidents and Modifications

Since we were very realistic when we did the first planning of the project, there haven't been many notorious modifications in the time plan. Most of the little modifications have been enlarging the time for each task, and overlapping it more with the next one in order to have time to control and test if the previous task is correct and there are not any bugs or unexpected problems. Another modification done is that we needed more time to acquire all the data for the *PulseID* dataset.

# Chapter 2

# State of the art

## 2.1. Biometric Pulse Identification

Most of the approaches in the literature for biometric pulse identification rely both on involving Electrocardiography (ECG), based on the electrical activity of the heart, and on a carefully design, segmentation and extraction of expert features from the pulse signal (Israel et al., 2005; da Silva et al., 2013). A decoupled approach which comprises mainly two stages is usually described (Gu et al., 2003; Übeyli et al., 2010; Choudhary and Manikandan, 2016).

Firstly, biomarkers or features are extracted from the pulse ECG or Photoplethysmography (PPG) signals, also known as front-end processing. Then, template feature vectors feed a second stage that performs model learning. Nonetheless, such features are designed by hand and strongly depend on a high expertise both on the knowledge of the addressed task and on acquisition nature of the pulse signal itself. For instance, in (Gu et al., 2003) an experiment on a group of 17 subjects was performed, where the authors studied four time domain characteristics, as time intervals, peaks and slopes from the PPG signals reporting successful accuracy rates of 94% for human verification. In the work of (Übeyli et al., 2010), feature extraction on the PPG, ECG, Electroencephalography (EEG) signals was performed based on eigenvector methods. Spachos et al. (2011) studied four feature parameters, peak number, time interval, upward slope and downward slope. The study from (Kavsaoğlu et al., 2014) is intended for exploring the time domain features acquired from its first and second derivatives, where a group of 40 features were extracted and ranked based on a $k$-nearest neighbor algorithm. Choudhary and Manikandan (2016) perform a comparison of three methods based and proposed the pulsatile beat-by-beat correlation analysis, the rejection or acceptance of subject is performed based on the maximum similarity.

Finally, more recent works (Jindal et al., 2016) make use of deep belief networks and Restricted Boltzman Machines as classifiers. With the advent of *deep neural network* architectures (DNN), such as convolutional based neurons, end-to-end processing pipelines are gaining popularity by building architectures capable of learning features directly from raw data. For instance, in computer vision (Le, 2013) or speech processing (Segura et al., 2016; Gong and Poellabauer, 2018) novel feature learning techniques are applied directly on the raw representations of images and audio, avoiding the signal parameterization or any other prior preprocessing.

## 2.2. Deep Learning

*Deep learning* (DL) has become a household name since its popularity and appearances in the mass-media have risen in the last years. But the truth is that DL is not that recent. The reason that it only appears to be new is that it was rather unpopular for several years and it has gone through many different names. It was born under the name *cybernetics* in the 1940s changing to *connectionism* in the 1980s until the current deep learning.

The first thing to clarify about deep learning is its relation with other popular names like *artificial intelligence* (AI) and *machine learning* (ML). As it can be seen in Figure 2.1, DL is a kind of *representation learning* (RL) which is in turn a kind of machine learning (ML), used from many but not all approaches to artificial intelligence (AI). The earliest algorithms of deep learning aimed to be computational models of *biological learning* (how learning happen or could happen in the brain) motivated by two ideas:

- ◇ Proof by example behaviour is possible and intelligent.
- ◇ Intelligence can be built by doing reverse engineering the computational principles behind the brain and replicating its functionality.

On the other hand, the modern perception goes beyond this approach and appeals to a more general principle of learning multiple levels of composition, learning and adapting more complex structures and patterns.

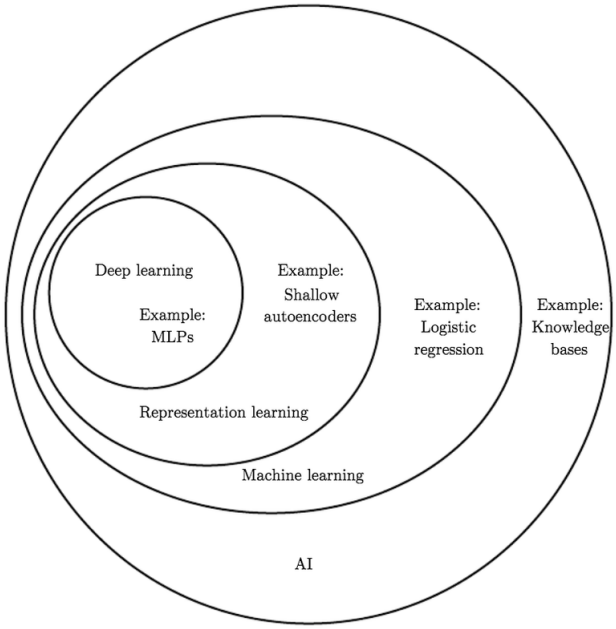



Figure 2.1: Venn diagram showing the relation between DL, RL, ML and AI. Extracted from (Goodfellow et al., 2016)

The classic machine learning techniques depended on the processing skills of the researcher for a specific type of data. In order to build a reliable and useful algorithm, the researcher had to be an expert in the field he was exploring. In other words, the analyst must had to be able to design a feature extractor that serialized the raw data, such as the samples in audio or pixels in images, into another structure from which the algorithm could detect, extract and classify patterns. However, deep learning methods are able to learn multiple complex representations within different levels and depths by composing non-linear models. Each non-linear model transforms the previous one, combined or not with the raw input data, into a more abstract representation. Thus, the complexity of the functions learned is bigger than the complexity of the functions learnt with linear models and grows as new transforms and combinations between non-linear models appear. Therefore, there are no hand-crafted features, they are learnt directly from the data, which represents the biggest breakthrough of deep learning.

So, since it is not necessary anymore to be an expert in the field you are analysing/studying, more people investigated more fields contributing to the evolution of the state-of-the-art in many fields and nowadays it is difficult to find a field not explored by DL.

### 2.2.1. Neural Networks

Neural Networks (NNs) are the basic architecture and form every deep learning algorithm. They are defined by neurons, a basic unit that performs a combination

of linear and non-linear operations, see Figure 2.2 (a). Understanding what a neuron is and familiarizing how it works is important in order to comprehend the Neural Networks.

An input vector $\boldsymbol{x} = \{x_0, x_1, x_2, \ldots, x_N\}$ is injected into the neuron and it computes an output. Each input has an associated weight $w$ according to each input importance. The output results from summing up all this weighted inputs $\boldsymbol{w}^T\boldsymbol{x}$ with a bias $b$ term, which provides every node with a trainable constant value.

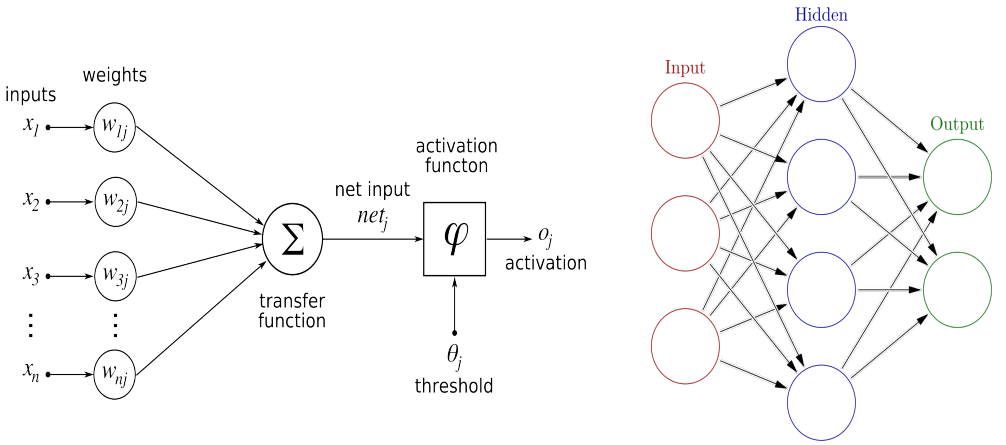
$$y = \boldsymbol{w}^T\boldsymbol{x} + b \tag{2.1}$$

The result of this linear operation is passed through a function $f$ (see Equation 2.2) called activation function. It maps the resulting values in between 0 to 1 or $-1$ to 1, etc. depending upon the function. It is usually exemplified with the *sigmoid* function, for two-class logistic regression. Since probability of anything exists only between the range of $[0, 1]$, sigmoid is the right choice. However, it is not the only non linear function that can be applied and other functions can be valid in other regressions.

$$o = f(y) = \frac{1}{1 + e^{-\theta y}} \tag{2.2}$$

So, a NN is created by connecting and stacking many of these simple neurons so that the output of a neuron can be the input of another. We call *layer* to a collection of nodes (neurons) operating together at a specific depth within a neural network. Every NN, even the simplest, has three types of layers. The first and the last layer are called input layer and output layer, logically. The layers between them are called hidden layers, in which each layer can apply any function you want to the previous layer (usually a linear transformation followed by a squashing nonlinearity). Hence, the hidden layers' job is to transform the inputs into something that the output layer can use. This hierarchy increases the complexity and abstraction at each level, and it is the key that makes deep learning networks capable of handling very large data sets. Figure 2.2 shows a schematic representation of one artificial neuron and a representation of a basic NN.

NNs learn from examples, like people. This makes very understandable the learning process of a neural network. Let's suppose you want to build a system capable of verifying one person's identity using the fingerprint. Firstly you need to collect a large dataset of fingerprints that includes genuine fingerprints (from the subject we want to authenticate) and impostor fingerprints (from as many different people as we can find). The first group compose the target data and the second group the impostor or world




(a) Artificial neuron model. Reprinted from (Chrislb, 2005)

(b) Basic neural network architecture. Reprinted from (Glosser.ca, 2013)

Figure 2.2: Basic artificial neuron (a) and neural network (b) architectures

data. During training, the machine sees pairs of fingerprints and labels, which indicate whether that fingerprint is from target data or impostor data. The system tries to find out what have the genuine fingerprints in common and produces one output as a result of the operations carried out by all the neurons: *true* '1' (if the data comes from the genuine user) or *false* '0' (if the data comes from an impostor user). This scores can be interpreted as the probabilities of the input fingerprint to be genuine or not. Then, the activation function, sigmoid in this case, is applied in order to squash this scores to be between 0 and 1 and divide each output such that the total sum of the outputs is equal to 1. Then, the system checks these probability scores for each class with the input label and modifies its internal adjustable parameters (the weights of each neuron) to maximize the accuracy in the class predictions. After training, the system performance is measured on a different, unseen data set called test. With this, we are testing the machine's ability to produce good results on new inputs that have not been used during the training process.

Neural networks are typically feedforward networks in which data flows from the input layer to the output layer without looping back. But for systems with sequential entries, such as audio or images, there are architectures where the inputs cross the network in different ways, to take advantage of the previous entries, such as *convolutional neural networks* (CNNs), which will be explained in the following section, or *recurrent neural networks* (RNNs).

### 2.2.2. Convolutional Neural Networks

Convolutional neural networks (CNNs) (LeCun et al., 1989) are a specialized kind of NN for processing data and specially useful when it comes to identify patterns or objects.

A CNN is made up of several layers that process and transform an input to produce an output. They are widely used in computer vision, mainly in face recognition, scene labelling, image classification, action recognition and document analysis. But also the fields of speech recognition and text classification for natural language processing. Although we use 1–D CNN in this thesis, the following explanation uses 2–D CNN due to its better representation.

In order to understand how CNNs work, it is important to be familiarised with this three concepts: local receptive fields, shared weights and biases, and activation and pooling.

In a typical neural network, each neuron in the input layer is connected to a neuron in the hidden layer. However, in a CNN, only a small region of input layer neurons connect to neurons in the hidden layer. These regions are referred to as local receptive fields (see Figure 2.3). The local receptive field is translated across an image to create a feature map from the input layer to the hidden layer neurons. The way to do it efficiently is by using the convolution.

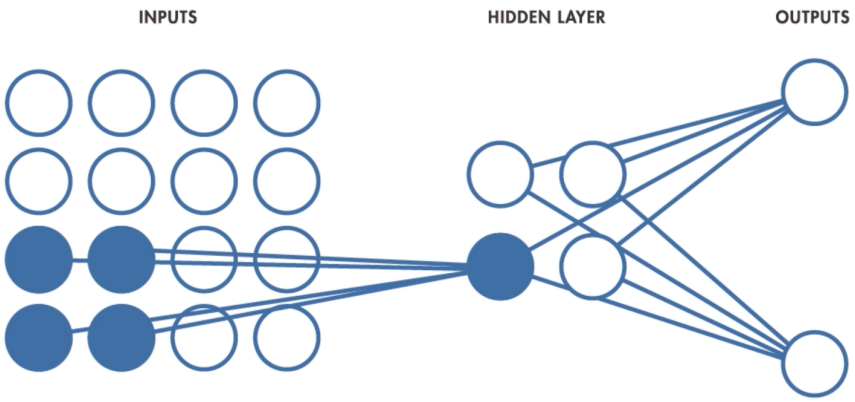



Figure 2.3: CNN shared weights and biases. Note that not all input nodes are connected with all nodes in the hidden layer. Extracted from (S. Patel, J. Pingel, 2017)

CNNs have neurons with weights and biases. The model learns these values during the training process, and it continually updates them with each new training example. However, in the case of CNNs, the weight and bias values are shared between the *kernels* that form the layers. This means that all the hidden neurons are detecting

the same feature (p.e. an edge) in different regions of the image. In 1–D CNNs, the hidden neurons would detect peaks, noise or other patterns in different chunks of the signal. This makes the network more robust and it will be able to identify the feature whatever it is in the image.

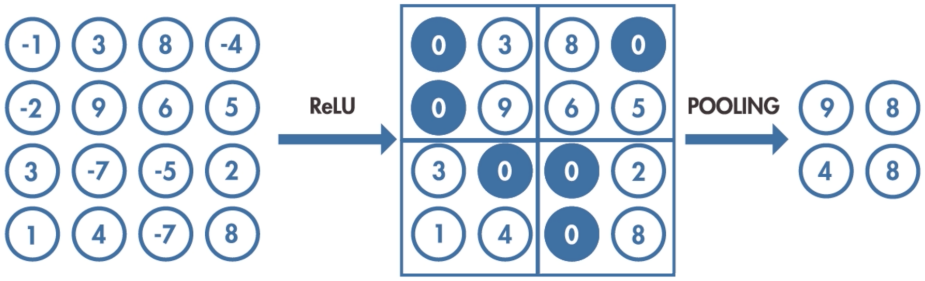



Figure 2.4: CNN ReLU + pooling operations. CNNs reduce dimentionality and with this, the number of features to learn. Extracted from (S. Patel, J. Pingel, 2017)

The activation step applies a transformation to the output of each neuron by using activation functions. *Rectified Linear Unit*, or ReLU, is an example of a commonly used activation function. It takes the output and if it is positive, the function remains it the same value but, if the output is negative, the function maps it to zero. The output of the activation step can be transformed by applying a pooling step. Pooling reduces the dimension of the feature map by condensing the output of small regions of neurons into a single output. This helps simplify the following layers and reduces the number of parameters that the model needs to learn.

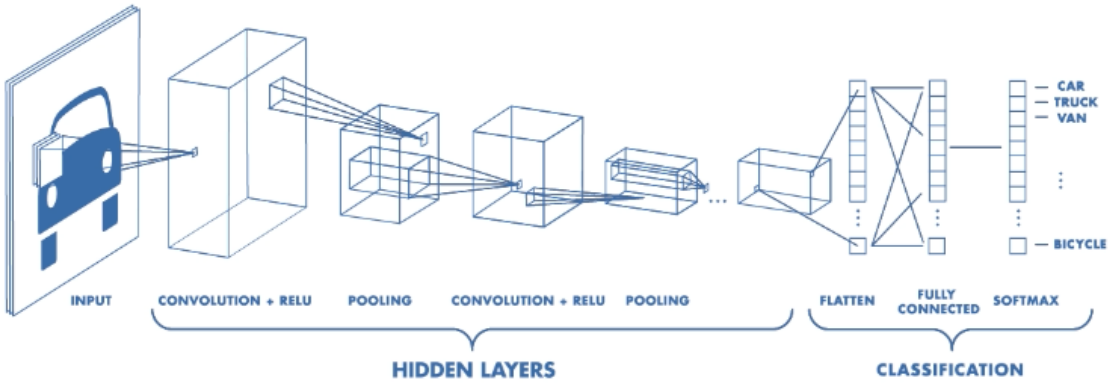



Figure 2.5: CNN classic architecture for a content-based image retrieval. Extracted from (S. Patel, J. Pingel, 2017)

The Figure 2.5 shows a basic convolutional neural network configuration for an object detection system. The hidden layers contain different filters of different lengths

that aim to find patterns at various depths. Then, a flatten vector displays all features in one single vector and all features in it can be fully-connected with a dense layer. This layer together with the softmax (in multi-class logistic regression) or the sigmoid (in two-class logistic regresison) perform the classification.

### 2.2.3. Optimization

Optimization refers to the task of either minimising or maximising some function $f(w)$, called objective function, cost function or loss function, by altering $w$. We can reduce $f(w)$ by moving $w$ in small steps in the opposite direction of the derivative $f'(w)$. This method (Cauchy, 1847) — which computes the cost and the gradient for all the training data — is called gradient descent and the *Stochastic Gradient Descent* (SGD) based on it — which computes the gradient and the cost from different groups of few samples called *batches* — is widely used in deep learning since it often performs very successfully. The use of SGD in the neural network settings is motivated by the high cost of running back propagation over the full training set. SGD can overcome this cost and still lead to fast convergence. So, in order to minimise $f$, we would like to find the direction in which $f$ decreases the fastest. To do so, we use the directional derivative in the steepest descent method.

$$w' = w - \alpha \nabla_w f(w) \tag{2.3}$$

where $\alpha$ is the learning rate, a positive scalar that indicates the size of the step. It is important to set up a good learning rate because if it is too small, it will take too long to reach the convergence, but if it is too high, maybe the convergence is never achieved. $-\nabla_w$ represents the negative gradient of the function we want to optimize and indicates the direction in which $x$ has to move. Like every project that has to start, it is important to know about in which point is the state-of-the-art in order to decide the perspective from which to address the hypothesis. In this chapter we have reviewed the state of the art in biometric pulse identification and deep learning. Passing through its classic architectures and recent applications. The components exposed here are essential to understand our end-to-end solution since they are the foundation above this project is based on. Our solution is explained in Chapter 4 but first, in next chapter the datasets that have been used are depicted.

# Chapter 3

# Datasets

Due to the unprecedented character of this work, the techniques that would be used must be avant-gardist and never used before together with PPG. So in order to fulfil these expectations, deep learning was a strong point and must be in the project, which conditioned all the methodology.

As it is mentioned in Chapter 2, deep learing requires a lot of data to be able to build models and it will be easier for the algorithm to do it from a specific-oriented dataset. We regarded that there is no publicly available dataset created for the specific purpose of biometric identification. The only dataset that we could use is the *Troika* dataset, detailed in Section 3.2.2, that is oriented for a Heart Rate tracking in noisy conditions, but all acquisitions are from different subjects. Nevertheless, this supposes that the signal is very noisy and supposes an extra problem for the algorithm to identify subject-unique patterns in the data. Thus, a specific dataset called *PulseID* (Section 3.2.1) had to be created in order to have reliable data. So the first thing necessary was to develop a prototype.

## 3.1. Prototype

The prototype must fulfil these specifications:

- ◇ As similar to a wearable bracelet as possible. Accordingly, the PPG sensor must be of the same type than the ones used in, p.e. activity tracker bands.
- ◇ Open-sourced. In order to make it the most adaptable and flexible.
- ◇ Minimum cost. That is one of the reasons for using PPG sensors intead of ECG sensors and the prototype must continue this purpose.

Following the mentioned specifications, the most suitable way to build a prototype is by using a Raspberry Pi (Upton, 2018) as a computational module due to its reduced dimmensions, the free license and its high compatibility to work with other modules. Besides the Raspberry Pi, the pulsesensor (Murphy and Gitman, 2018) was mounted for obtaining the PPG signals and finally, we decided to use the 10 bits Analog to Digital Converter (ADC) MCP3308 (Microchip, 2008) to transform voltage variations into digital samples. This setup is ridiculously cheap, as it can be seen in Table 6.3.

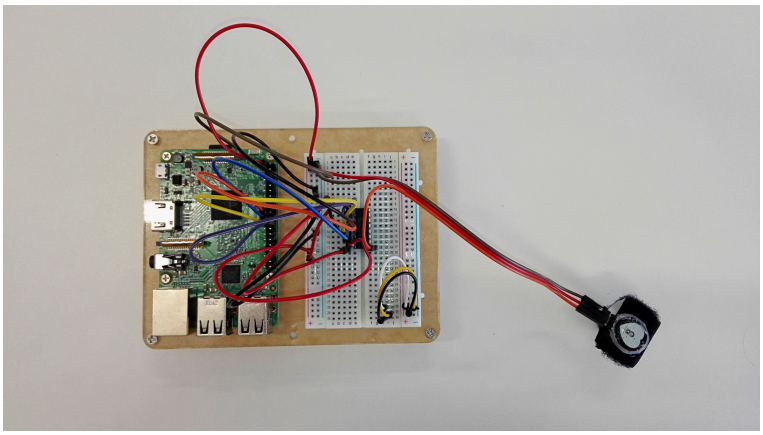


Figure 3.1: Prototype setup. From left to right: Raspberry Pi 3, ADC MCP3008, Pulsesensor. Note that a breadboard and jumper wires are used as a connection platform

## 3.2. Datasets

Two different datasets are employed in this work to conduct person verification experiments through PPG signals. Firstly, a new corpus was collected aiming to fulfil the need of, to the best of authors' knowledge, a public domain dataset specifically created for PPG biometric identification. For such a purpose, a new PPG dataset has been collected, named as *PulseID*[1], in a quiet office environment. Secondly, aiming to verify the robustness of our approach in more challenging conditions, the Troika (Zhang et al., 2015) dataset is used. In contrast, Troika recordings are acquired for subjects on a treadmill, walking and running at different speeds.

Table 3.1: Summary stats for both databases, Troika and PulseID. The "duration" column stands for the average duration in seconds of the total acquired samples per subject

| Dataset | Subjects | Gender (m/f) | Duration |
|---|---|---|---|
| PulseID | 43 | 31/12 | 240s. |
| Troika | 20 | 20/— | 317s. |

[1]The *PulseID* dataset is available upon request from the authors and agreement of EULA for research purposes.

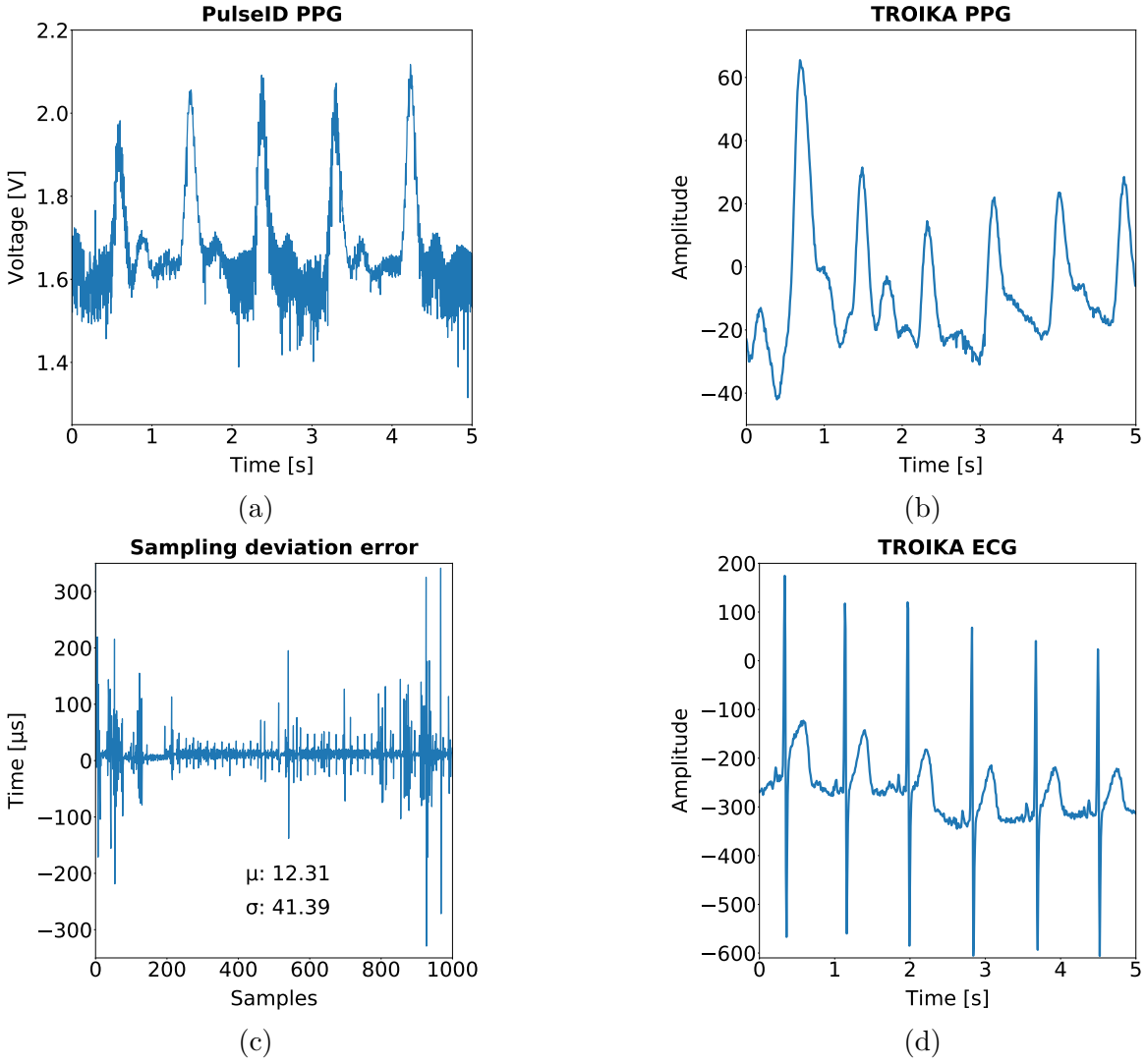



Figure 3.2: Five seconds PPG excerpts from PulseID (a) and Troika (b) databases. (c) Time sampling deviation in $\mu$ seconds from the same PPG signal in (a). In (d), a five seconds ECG excerpt from Troika

### 3.2.1. PulseID

For the PulseID data acquisition, the pulse sensor described in (Murphy and Gitman, 2018) is employed. It is essentially a photoplethysmograph, a well known medical device used for non-invasive heart rate monitoring, consisting of a green LED and a photo-detector. The heart pulse signal that comes out of the pulse sensor is an analog fluctuation in voltage, with associated waveform known as photoplethysmogram or PPG, see figures 3.2 (a) and (b). The pulse sensor responds to relative changes in illuminance. For a sensor placed in the subject's skin, the reflected light back to the photo-detector changes during each pulse due to blood flowing what is perceived as variations in the voltage signal.

The process of data acquisition involved 43 volunteers (31 male and 12 female) with ages ranging from 22 to 55. Subjects were seated down in a calmed, relaxed

and quiet office environment while the recordings. The PPG sensor was attached to the fingertip of the right index finger by a belt. PPG acquisitions, lasting roughly one minute, were recorded from each subject and repeated 5 times along the same session. PPG analog signal was sampled at 200 Hz rate. For doing so, a python code was developed aiming to perform sampling synchronization with a voice recorder (data used for another projects) while reading from the ADC and ensuring a tolerant averaged sampling rate deviation of $\mu = 13.32$ $\mu$s and $\sigma = 202.58$ $\mu$s per subject, see plot (c) in the Figure 3.2 as an example. For comparison purposes, an ECG waveform from Troika is depicted in Fig. 3.2 (d). It is worth noting that no pre-processing is performed to the raw PPG acquired signals. It can be seen in the higher noise levels present in the acquired PPGs, where various artifacts are expected to be found. For instance caused by analog circuit noises or medium illuminance changes, respiration or base deviation arising from movement.

### 3.2.2. TROIKA

In order to verify the robustness of our approach, the Troika dataset (Zhang et al., 2015) is used. It consists of an ECG signal, two-channel PPG signals and three-axis acceleration signals from 20 male subjects with ages ranging from 18 to 35. For each subject, the PPG signals were recorded from wrist by two pulse oximeters with green LEDs of wavelength 609 nm. To make the data recordings similar to practical world readings, the pulse oximeter and the accelerometer were embedded in a wristband and all signals were sampled at 125 Hz. During data recording, subjects walked and ran on a treadmill at different speeds for an average time of 4 minutes[2] per subject.

The are 2 different sequences detailed in Table 3.2: *TYPE_01* sequence consists in: 2 km/h for 0.5 minutes, 8 km/h for 1 minute, 15 km/h for 1 minute, 8 km/h for 1 minute, 15 km/h for 1 minute, and 2 km/h for 0.5 minutes. And the *TYPE_02*, which consists in resting for 0.5 minutes, 6 km/h for 1 minute, 12 km/h for 1 minute, 6 km/h for 1 minute, 12 km/h for 1 minute, and resting again for 0.5 minutes.

Table 3.2: Troika acquisitions structure

| **Sequence** | 30” | 1’ | 1’ | 1’ | 1’ | 30” |
|---|---|---|---|---|---|---|
| TYPE_01 | 2 km/h | 8 km/h | 15 km/h | 8 km/h | 15 km/h | 2 km/h |
| TYPE_02 | 0 km/h | 6 km/h | 12 km/h | 6 km/h | 12 km/h | 0 km/h |

In DL data is very relevant so it is important to ensure we have good data to develop a system from. So, once we have all the necessary data, we can start developing the system.

[2]The reality is that not all samples have the same duration so, is prefered to talk in terms of average time per user.

# Chapter 4

# End-to-end biomarker learning

This chapter embraces all processes, tasks, thoughts and discussions that have brought this project to success. Starting in the documentation process and ending in the demonstration prototype, passing through the data collection, the development code and validation metrics.

## 4.1. Experiment Design

Once all data has been collected it is time to design the experiments in order to find and demonstrate the best architecture and the best algorithm that brings with the best results.

The end-to-end authentication system is validated and tested averaging the performance of 31 subjects for training and 12 impostor subjects for the final test. It sums up a total of 43 subjects for the PulseID database. In the case of Troika, 15 training subjects and 5 impostors for the final test accounting for a total of 20 subjects were used. So the total number of enrolled subjects is 31 and 15, respectively. For each user, the system is trained with the training data (31 subjects in case of PulseID) in order to find the best architecture of the system. The best system is the one that have a better performance in average for all the training subjects. After the best system is chosen, the performance of the system is computed with the data reserved for the final test.

Then, it is necessary to manage and distribute the data available. Since we want a fair results — that not only are not the consequence of a overfitted system, but that represents a real scenario — we reserve some data from the target subject and all data from a few impostors data. That will simulate a real scenario in which a new impostor

not seen by the system tries to authenticate as the genuine user.

Another objective to take into consideration is the fact that the final product is a biometric authentication system, so it is fundamental that the system is able to verify one person's identity within a reasonable time. That means that if the system specification (see Section 1.2) is ability to authenticate in 3 seconds, the system has to be trained with chunks of 3 seconds. In an attempt to reach the systems limit, it is trained with 1 second signal blocks. Note that in a 60 bpm (beats per minute) Heart Rate there is one beat each second, there is no data-partition preprocess to ensure that a full peak is taken because every chunk contains a different signal, unless the subject Heart Rate is stable at 60 bmp during all the acquisition, which is highly improvable.

As it can be seen in Table 4.1, data is distributed in 4 groups: *Train*, *Validation*, *Develop* and *Test*. The data from the first 3 groups (training data) — that is, *Train*, *Validation* and *Develop* — is from the same subjects. This data is used to define and fit the neural network (NN) detailed in Section 4.1.1. The *Train* data is from where the PPG features are extracted and all distinctive characteristics are discovered. A combination of pairs (target, impostor) from the *Train* and *Validation* sets are used for network training and validation, performing parameter updating based on binary cross-entropy loss computed on the *Validation* set. The unusual *Develop* group, corresponds to the amount of data used by the authors to test possible network architectures as well as different parameters in order to obtain the best system setup. It includes the threshold selection. Finally, the *Test* set composed of unseen impostor subjects is employed for the fair assessment of the end-to-end proposed approach. This partitioning model prevents biasing and is an approach to a real scenario, in which it is impossible to have all impostors' PPG signals.

Table 4.1: Partition data for the different sets of PulseID dataset. Trials are expressed in seconds of signal and averaged per subject. Since the trial size of the experiments showed is one second, the number of *Target* and *Impostor* data corresponds to number of trials or seconds

| **Dataset** | **Label** | **Train** | **Validation** | **Develop** | **Test** |
|---|---|---|---|---|---|
| PulseID | Target | 135 | 45 | 30 | 30 |
| | Impostor | 5,220 | 1,740 | 1,890 | 2,880 |
| | #Subjects | | 31 | | 12 |
| Troika | Target | 144 | 80 | 48 | 48 |
| | Impostor | 2,014 | 1,119 | 1,343 | 1,545 |
| | #Subjects | | 15 | | 5 |

### 4.1.1. Algorithm & Architecture

As it is mentioned before, the system algorithm and the models training as well as the NN fine-tuning, is performed with the absolute transparency and strictness with the intention to reproduce a real scenario.

The first thing is to split the data as the Table 4.1 shows. This basic partition is done without any peak tracking preprocess and its saved into a *Pandas Dataframe* (Jones et al., 2001–). Once the initial data partition is done, it takes place what we called *random shuffles* creation. The idea behind random shuffles is to create balanced batches — in a balanced batch, there are the same amount of target samples than impostor samples — to use as *Train* batches in the NN. The intention is to have different batches formed by the same target data but different impostors data. Hence, the Random Shuffles are constituted by all *Train* data and then, random data excerpt from the pool which contains all the impostors data labelled as *Train*. Now this random shuffled batch is ready to be the input of the convolutional neural network (CNN) detailed in Figure 4.1.

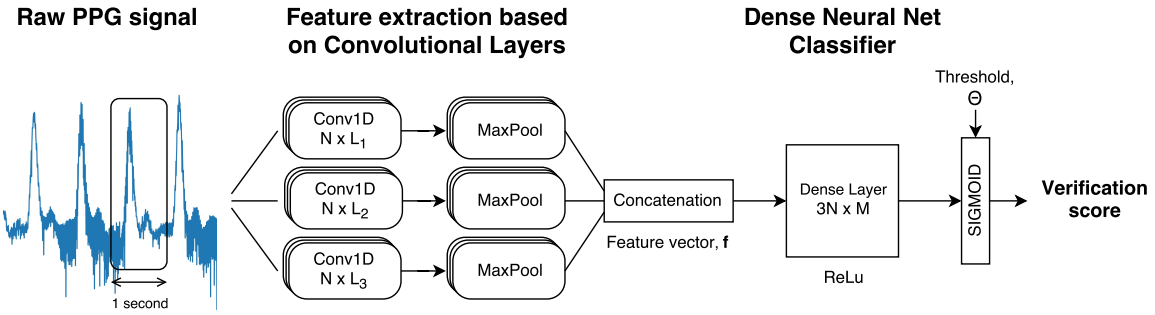



Figure 4.1: Proposed convolutional neural network architecture for end-to-end user verification using raw PPG signals. First, the raw signal is filtered by three parallel 1-D convolutional layers composed of $N$ filters of lengths $L_{1,2,3}$ followed by a global max-pooling operation. The resulting $3N$ features are then concatenated into the feature vector $\mathbf{f}$, which is used to perform the classification using a dense layer of dimensions $3NxM$ and the final layer of 1 output. ReLU activation function is used across all layers but the output layer, where a sigmoid activation is used to predict the verification score.

Let's assume the PPG signal input to the CNN is a vector, $\boldsymbol{x}$, whose elements are raw PPG samples $\boldsymbol{x} = [x_k, x_{k+1} \dots x_{k+K}]$ where $x_k$ is the PPG sample shifted by a given stride. In this work we used a value of 1 for time shifting and the $x_k$ sample with a fixed size from 1 second at 200 samples per second.

The activations at the first convolutional layer comprise $N = 6$ filters and we denoted them as $\boldsymbol{h}_n = [h_1 \quad h_2 \cdots h_N]$. Therefore, the convolutional layer operation can be seen as a convolutional operation of each filter on the input raw PPG,

$$h_n = \theta\left(\boldsymbol{w}_n \boldsymbol{x}^T + b_n\right), \tag{4.1}$$

where $\theta(x)$ is the activation function corresponding to Rectified Linear Units (ReLU) (Section. 2.2.2), $\boldsymbol{w_n}$ is the weighting vector and $b_n$ the bias term for filter $h_n$. Following the convolutional filters, max-pooling layers perform local temporal max operations over the input sequence, selecting the maximum in a window of $d$ size. More formally, the transformation at starting sample vector $n$, $\boldsymbol{c}_k^n$, corresponding to the filter output sequence of the first convolutional layer and $j$th filter is:

$$\max \quad c_s^n \qquad k - \tfrac{(d-1)}{2} \leq s \leq k + \tfrac{(d-1)}{2} \tag{4.2}$$

The pooling operation compacts even more the original signal by computing some stats, commonly such as maximum, mean and variance, from the CNN output. Note that the CNN-maxpooling feature learning architecture applies 1-dimensional convolutions and pooling operations performed along the time axis as previous works in (Segura et al., 2016). For this work maximum pooling is used by selection of the maximum values from the CNN filter outputs. Next, a flattering operation is performed, see Figure 4.1, that aims at stacking together all the CNNs outputs, creating a feature vector ready to be presented to the network classifier.

In overall, the end-to-end architecture comprises a total of 4 layers. In the input, a convolutional layer with different amounts of filters and lengths (see Figure 4.1) followed by a max pooling layer. At the back-end, a fully connected neural net composed of 2 layers with 256 units. ReLu units are employed in all layers, including CNNs, except for the output layer of the dense network, where we used a sigmoidal unit. The dense layer is employed as a back-end for the modeling of the salient features computed by previous convolutional steps and a Sigmoid activaction function (Eq. 4.3) used in the output layer. It is worth to note that no dropout is used during network training.

$$S(x) = \frac{1}{1 + e^{-\theta x}} \tag{4.3}$$

The framework (Cortès et al., 2018) has been developed in Keras (Chollet et al., 2015) and using Tensorflow (Abadi et al., 2015) as back-end. We do not perform an exhaustive search of network parameters and we restrict experiments by using few learned biomarkers. For instance, we compute 15 features before the dense layer for the reported 1 second experiments, see vector $\mathbf{f}$ in Figure 4.1. The network is trained using Stochastic Gradient Descent (SGD) attending to binary cross-entropy as a loss

function and accuracy as a metric, with mini batches of size 270 composed of 135 target trials and 135 impostor ones. Given a *Train* set of PPG excerpts from a subject, at each training mini batch, the impostor samples are randomly picked up from the available pool of impostor chunks so in each training iteration, new impostor data is seen as an intention to maximise the variability, see table 4.1. An early stopping criteria is also defined in order to speed up the training, yielding in most of the cases to few tens of training mini batches before reaching *patience* steps.

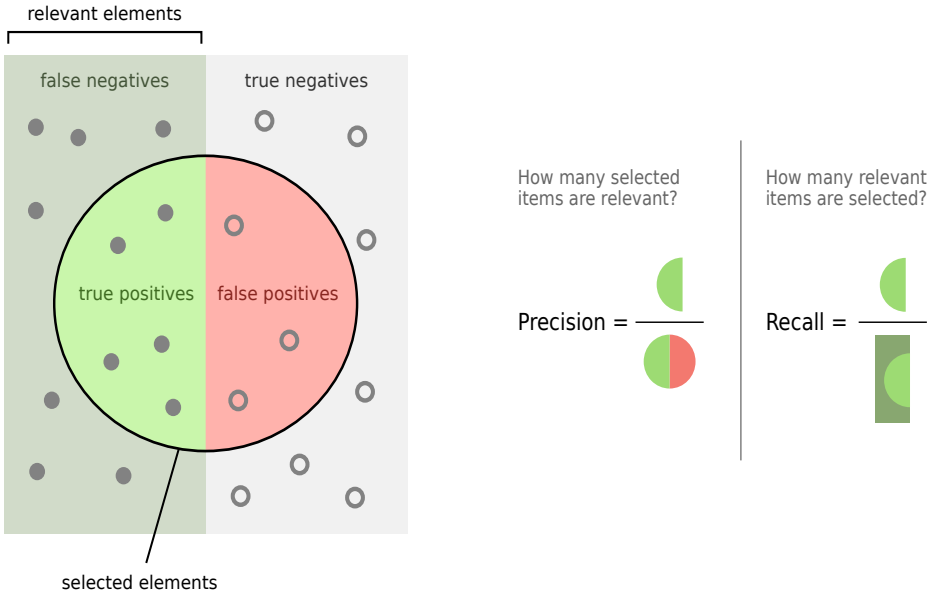
## 4.2. Metrics

In binary classification there are only 2 types of real values: Positive values '1' and negative values '0'; therefore there are two possible prediction outcomes: 1 and 0. The Table 4.2 is a graphic representation of a generic confusion matrix.

Table 4.2: Confusion Matrix Table

| | | **Prediction outcome** | | |
|---|---|---|---|---|
| | | 1 | 0 | |
| **True value** | 1 | True Positive | False Negative | P |
| | 0 | False Positive | True Negative | N |
| | | P' | N' | |

The confusion matrix is very useful because its easy to see which are the main errors in the predictions. Then, the system can be reconfigured and with this, increasing the accuracy and the predictions and also, the performance in the Precision-Recall curve, explained below.




(a) Elements classification (b) Precision and Recall visualisation

Figure 4.2: Precision and Recall representation. Adapted from (Walber, 2014)

Precision and recall are measures of relevance and give us an accurate evaluation of the performance of a system. The *precision* is the answer to the question: From the elements predicted as positive, how many of them are really positive? And the *recall* answers the question: From all the elements that are really positive, how many of them are predicted as positive? We can express it with the following equations using the notation of Table 4.2:

$$Precision = \frac{TP}{P'} = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{4.4}$$

One of the most used metrics when it comes to evaluate systems is the ROC curve (Hanley and McNeil, 1982). It is very similar to the precision-recall (PR) curve and it represents the same since both contain the same points. The key difference between the two curves is that ROC curves will be the same no matter what the baseline probability is, but PR curves may be more useful in practice for needle-in-haystack type problems or problems where the *positive* class is more interesting than the negative class. So, in order to answer the question "How well can this classifier be expected to perform in general, at a variety of different baseline probabilities?" the best option is to use the ROC curve since it is a better metric for evaluating a performance in general (Davis and Goadrich, 2006).

So, in this thesis we choose the best configuration of the system regarding the ROC

plots of the experiments (see Chapter 5) and the best configuration corresponds to the curve with better Area Under the Curve (AUC).

Once the best configuration is chosen, it is necessary to establish the operation point that will determine the final performance by setting the *False Match Rate* (FMR) and the *False Non-Match Rate* (FNMR) metrics. The ones that the standard (International Organization for Standardization, 2006) in Biometrics establishes for the evaluation of binary classification systems.

So, the FMR and FNMR represent the errors in the predictions. The *False Match Rate* is related to security: the minimum FMR implies maximum security because that means that there is no impostor predicted as the genuine subject. Otherwise, the *False Non-Match Rate* is an indicator of usability: the minimum FNMR means maximum usability because every time a genuine subject is authenticated, the system will predict him as the genuine subject.

$$FMR = \frac{FP}{P' + N'} \qquad FNMR = \frac{FN}{P' + N'} \tag{4.5}$$

If we relate this equation with the confusion matrix in Table 4.2 we can come up to the conclusion that if maximum accuracy is desired, then its necessary to maximise the diagonal of *True positives* (TP) and *True Negatives* (TN). The difficulty comes in the decision of the threshold or operation point because, as it can be seen in Figure 5.3 there is a compromise between FMR and FNMR. If you enhance one, the other will be deteriorated.

## 4.3. Demonstration / Final configuration

The only necessary thing to do to test the system is to load the model of the user that will be the target user and define its fine-tuned threshold. With this, the system is ready to authenticate.

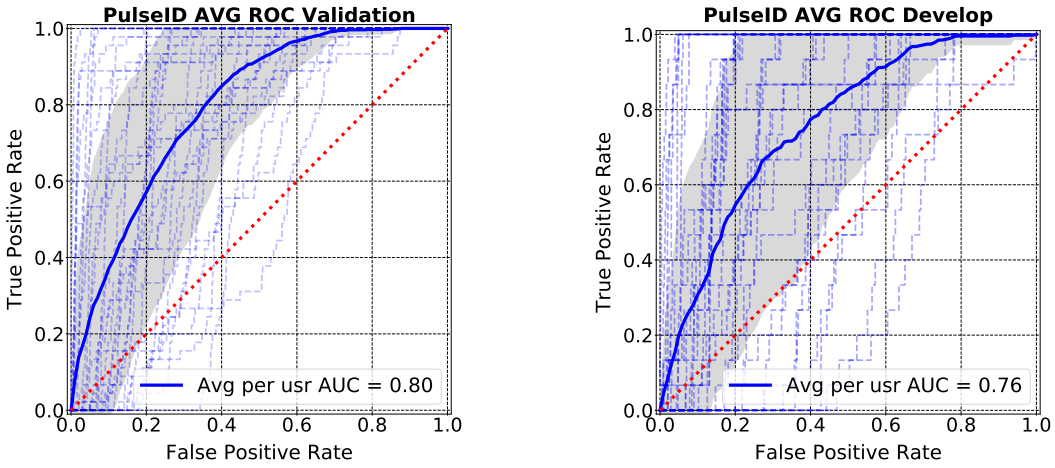
One way to increase the performance of the system and its robustness, losing a bit of a usability, is to combine $X$ verification attempts of 1 second into one bigger verification of $X$ seconds. With this the system would be more robust in front of False Matches, but it would also would reduce the probability of True Matches, so again, there is a compromise between usability and security. We propose evaluating the verification after 4 individual tries of 1 second. And considering genuine a candidate if the result in 3 out of the 4 experiments is favourable, or otherwise, treating the candidate as an impostor if in 2 or more experiments the conclusion is adverse.

# Chapter 5

# Results

This Chapter presents the results of the evaluation of the system proposed in Chapter 4. First, the evaluation of the configuration performances is shown. Then, once the best configuration is established, the operation point is discussed.

The Figure 5.1 shows the ROC curves, per *Validation* and *Develop* sets in PulseID and averaged per subjects, solid line, and its standard deviation, shadowed area. For the sake of comparison, the same curves are depicted in Figure 5.2 (b) per each dataset and partition.




(a) Average ROC for Validation set

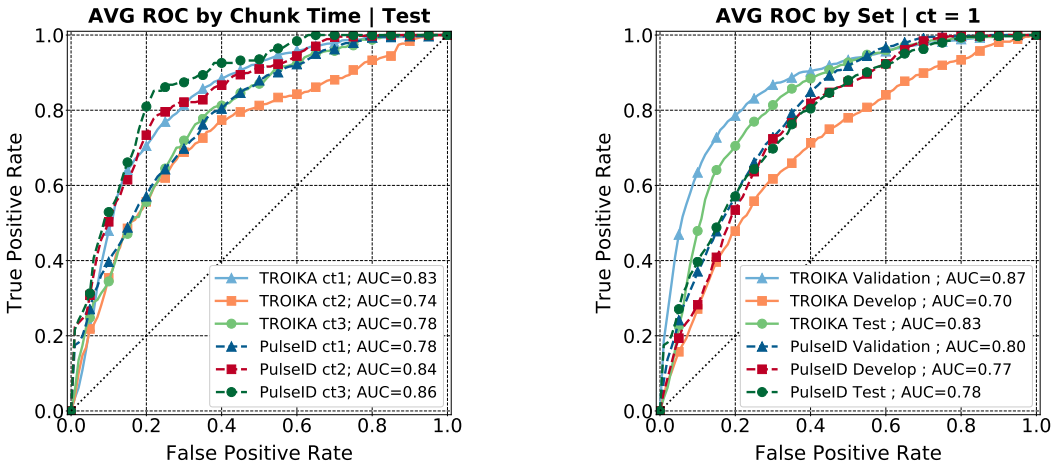
(b) Average ROC for Develop set

Figure 5.1: Average ROCs using $N = 6$ filters for each filter sizes of $L_{1,2,3} = 50, 30, 20$. The painted area corresponds to the area within the standard deviation of the AUC. Dashed lines stands for each subject AUC curves

Although an exhaustive search of the best network architectures or a fully tuning of parameters is not performed, we experiment with different window and filter sizes. For

the reported figures, we select $1, 2, 3$ second excerpts extracted from original raw PPG, homogeneously segmented and with no overlap for testing trials. The experiments are performed in PulseID data and best values, in terms of number of filters and size, are directly applied in Troika.

In overall, the results support the suitability of the end-to-end architecture in both datasets, although as observed in ROC curves Figure 5.1, some subject's AUC present a not satisfactory behaviour suggesting more experimentation to understand possibles sources of such variability.




(a) ROC comparison according to different chunk times ct=1, 2, 3

(b) ROC comparison according to the set of data

Figure 5.2: Average ROC performance comparison. The curves have been computed as in Figure 5.1, averaged for all the subjects

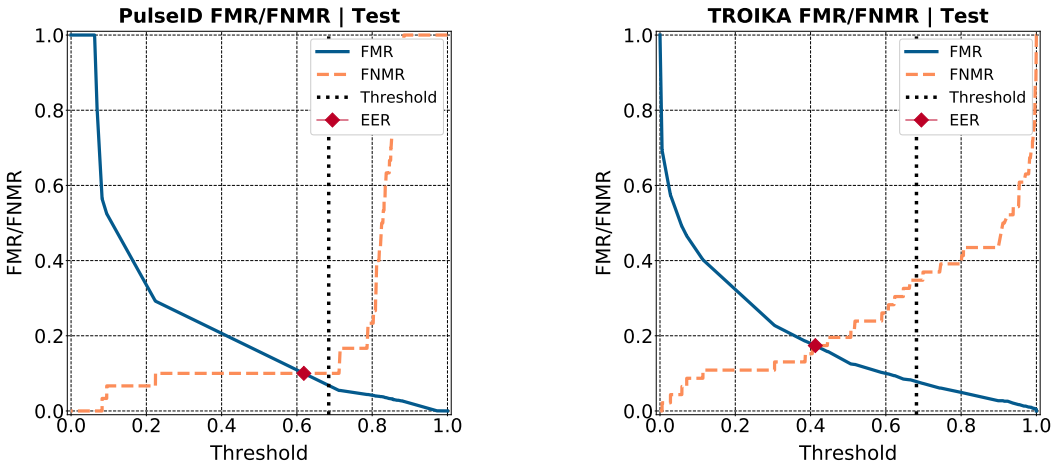
Homogeneous segmentation of the input PPG likely degrades system performance due to few samples are taken into account for training, see Table 4.1. However, it could be easily bypassed, e.g., by a randomly picking of excerpts thus increasing samples and segmentation variability in train and test.

The Figure 5.2 (a) and Table 5.1 report on the system performance for different trial sizes, ranging from 1 to 3 seconds. We can observe the generalization of validation results both in *Develop* and *Test* sets, showing high AUC values even for 1 second trial condition, of 0.78 and 0.83 per each dataset. Note the higher AUC degradation in Troika compared with the PPG data captured in the office condition and the AUC trend observed by increasing the *ct* time (chunk time), not observed in Troika likely due to motion artifacts. Obviously, the results in PulseID dataset, shows that is better to increase the ammount of signal used to authenticate but this also reduce the usability of the system and the scope of this thesis is to demonstrate that PPG signal can be used to authenticate people's identity.

Table 5.1: Average AUCs for all subjects within the same experiment: $N = 6$ filters for each filter sizes of $L_{1,2,3} = 50, 30, 20$. the ±variation corresponds to the AUC's standard deviation

| Dataset | Trial size | Validation | Develop | Test |
|---|---|---|---|---|
| PulseID | 1s. | 0.80±0.16 | 0.77±0.19 | 0.78±0.20 |
| | 2s. | 0.81±0.16 | 0.76±0.22 | 0.84±0.19 |
| | 3s. | 0.84±0.15 | 0.78±0.20 | **0.86±0.17** |
| Troika | 1s. | 0.87±0.09 | 0.70±0.16 | **0.83±0.12** |
| | 2s. | 0.73±0.30 | 0.66±0.21 | 0.74±0.24 |
| | 3s. | 0.85±0.14 | 0.71±0.16 | 0.78±0.18 |

Another parameter to determine in authentication systems is the operating point or decision threshold. It controls the trade-off between security, minimum FMR, and usability, minimum FNMR. Related to the algorithm complexity for trial decision, taking into account $L_i$ size of filters, pooling and dense layers operations, and 1s. trial lead to a number of Multiplier-Accumulator (MAC) operations (Sze et al., 2017) around $26K$, that translates into roughly 20ms for a Raspberry in order to perform person authentication every second.




(a) PulseID FMR/FNMR

(b) Troika FMR/FNMR

Figure 5.3: False Match Rate (FMR) and False Non-Match Rate (FNMR) ratio plots as function of threshold $\Theta$. The threshold line $\Theta$, see Fig. 4.1, corresponds to the operating point where the FMR is below 0.1 and the FNMR is minimum

Regarding the obtained results, our system has a good performance, reaching an AUC averaged among 31 subjects, of 86% in PulseID dataset and 83% in Troika dataset. Authenticating in 3 and 1 seconds, respectively. For a specific user, choosing the optimal operation point, we have a FMR below 0.1 and a FNMR around 0.6. These results reinforce the hypothesis that biometric authentication through PPG is possible and demonstrates that our system can be used as a security application although it is a feasibility study.

# Chapter 6

# Budget

This chapter details the costs involved in the development of the project. There are two main contributions to the cost of the project: labor, and computing and development equipment.

This project has been developed using the resources provided by *Telefónica I+D* (Telefónica I+D, 2018). Thus, the main costs of this projects comes from the salary of the researches, the time spent in it and the server time used for the training of the system. It is considered that the total duration of the project is 22 weeks, as depicted in the Gantt diagram in Figure 1.1.

### Labor Costs

Table 6.1 shows the contribution of labor to the cost of the project. We consider that my position is that of a undergraduate engineer, while the supervisor who was helping and advising me had a wage/hour of a senior engineer (for offering external services).

Table 6.1: Labor cost

| | Dedication | Wage / Hour | Total Hours | Total Cost |
|---|---|---|---|---|
| Undergraduate Engineer | 30 h/week | 8 EUR | 660 h | 5,280 EUR |
| Senior Engineer | 2 h/week | 100 EUR | 44 h | 4,400 EUR |
| | | | **TOTAL** | **9,680 EUR** |

### Equipment

Table 6.2 shows cost of the prototype. Prices were consulted at the official providers -https://www.raspberry.org/, -https://learn.adafruit.com/ and -https://pulsesensor.com/.

Table 6.2: Prototype cost

| Concept | Cost |
|---|---|
| Raspberry Pi 3 | 29.47 EUR |
| MCP 3008 | 3.22 EUR |
| Pulsesensor | 21.46 EUR |
| **TOTAL** | **54.15 EUR** |

Table 6.3 shows the contribution of computing and development equipment to the cost of the project. Prices were consulted at -https://www.avadirect.com/, -https://backmarket.es/ and -https://nvidia.com/. We chose a processing server as similar as possible to the one used throughout the project: 48-core Intel Xeon, 128 GB RAM, 4TB HDD storage.

All prices have been consulted in May, 2018. I find necessary to mention that the price of the graphic cards has been afected directly by the cryptocurrency phenomena. Another important point to mention is that it has been considered that the salvage value of the equipment is 0 EUR.

Table 6.3: Equipment cost

| Concept | Acquisition Cost | Scrap Value | Useful Life | Yearly Deprecation | Weekly Cost | Total Cost (22 weeks) |
|---|---|---|---|---|---|---|
| Processing server | 13,000 EUR | 5,000 EUR | 5 years | 1,600 EUR | 30.7 EUR | 676.93 EUR |
| Nvidia GTX 1080 Ti | 769 EUR | 230 EUR | 5 years | 107.8 EUR | 2.07 EUR | 45.6 EUR |
| Dell Latitude E5530 | 269 EUR | 80.7 EUR | 3 years | 62.77 EUR | 1.2 EUR | 26.5 EUR |
| **TOTAL** | 14,038 EUR | 5,310.7 EUR | 4.3 years | 1,770.57 EUR | 33.97 EUR | **749.03 EUR** |

### Total

The total cost of the project amounts to **10,483.18 EUR**.

# Chapter 7

# Conclusions

In this project an end-to-end architecture based on CNN is proposed to offer biometric authentication using learned biomarker directly from PPG raw signals and we reported evaluation results of the performance of our approach in two different datasets, Troika and PulseID.

Our end-to-end authentication approach and automatic learned biomarkers show a remarkable potential as authentication biometric method. Trial size dependent experiments, reported AUCs ranging $[78.2\%, 86.4\%]$ and $[73.8\%, 83.2\%]$, averaged among target subjects on PulseID and Troika datasets, respectively. Furthermore, the proposed system results in a low complexity that permits for continuous authentication in real-world scenarios.

The work reported accomplishes the requirements and specifications of Section 1.2 what suppose an absolute success. The system proposed is robust and usable (ability to verify in 1 second), with an accuracy over the 70% and with a low computational complexity that it can be embedded in a Raspberry, and all of this using deep learning techniques. And last but not least, a new dataset with 43 subjects has been recorded and its available to everyone for research purposes.

The Paper *End-to-end Photoplethysmography (PPG) Based Biometric Authentication by Using Convolutional Neural Networks* (Luque et al., 2018) is one of the results of this Thesis and it has been accepted in the 26th European Signal Processing Conference (EUSIPCO) that will be held in Rome, September 2018 [1]. It includes the dataset, the architecture and the results already exposed here.

Despite the good results, this thesis represents a feasibility study and future work is needed in case we want a reliable system that fulfils the strictest requirements of

[1]The Paper is attached to this Thesis in Appendix A

security systems.

In order to increase the usability of the system, one alternative to our solution could be substituting the convolutional neural network by a siamese network. With this, there would be a unique model instead of a personal model for each subject that has to be trained to be able to verify its subject identity. This unique model would be able to determine the similarity or not of two signals, in other words, if the first signal and the second belong to the same subject. A deep, conscious, sensible search of the best parameters and architecture would be also needed.

Another thing that is important to mention is that this thesis is based in some results of the experiments performed using two limited datasets. Enlarging the datasets is basic to get more consistent results.

We have published the code from this project (Cortès et al., 2018), as a contribution to the scientific community under the Apache License Version 2.0 (The Apache Software Foundation, 2018). The code and its requirements can be found at `-https://bitbucket.org/guillemcortes/pulseid-eusipco`. As it is mentioned in Section 3.2, The PulseID dataset is available upon request from the authors and agreement of EULA for research purposes.

# Bibliography

M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems. -https://www.tensorflow.org/, 2015. Software available from tensorflow.org.

Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

F. Chollet et al. Keras. -https://github.com/fchollet/keras, 2015. [Online; accessed 10-June-2018].

Tilendra Choudhary and M Sabarimalai Manikandan. Robust photoplethysmographic (ppg) based biometric authentication for wireless body area networks and m-health applications. In *Communication (NCC), 2016 Twenty Second National Conference on*, pages 1–6. IEEE, 2016.

Chrislb. Artificial Neuron Model. -https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png, July 2005. [Online; accessed 10-June-2018].

G. Cortès, J. Luque, J. Fabregat, and J. Esteban. PulseID Keras code for development and testing purposes. -https://guillemcortes@bitbucket.org/guillemcortes/pulseid-eusipco, 2018.

H. P. da Silva, A. Fred, A. Lourenço, and A. K. Jain. Finger ecg signal for user authentication: Usability and performance. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–8, Sept 2013.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Glosser.ca. Colored Neural Network. -https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg, February 2013. [Online; accessed 10-June-2018].

Y. Gong and C. Poellabauer. How do deep convolutional neural networks learn from raw audio waveforms? -https://openreview.net/forum?id=S1Ow_e-Rb, 2018. [Online; accessed 10-June-2018].

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. -http://www.deeplearningbook.org.

Y. Y. Gu, Y. Zhang, and Y. T. Zhang. A novel biometric approach in human verification by photoplethysmographic signals. *Proceedings of the IEEE/EMBS Region 8 International Conference on Information Technology Applications in Biomedicine, ITAB*, 2003-Janua:13–14, 2003.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

International Organization for Standardization. Information technology – Biometric performance testing and reporting. -https://www.iso.org/standard/41447.html, 2006. [Online; accessed 10-June-2018].

S. A. Israel et al. Ecg to identify individuals. *Pattern Recognition*, 38(1):133 – 142, 2005. ISSN 0031-3203.

V. Jindal, J. Birjandtalab, M. B. Pouyan, and M. Nourani. An adaptive deep learning approach for ppg-based identification. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6401–6404, Aug 2016.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. -"http://www.scipy.org/, 2001–. [Online; accessed 10-June-2018].

A. R. Kavsaoğlu, K. Polat, and M. R. Bozkurt. A novel feature ranking algorithm for biometric recognition with ppg signals. *Computers in Biology and Medicine*, 49 (Supplement C):1 – 14, 2014. ISSN 0010-4825.

Q. V. Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598, May 2013.

Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, pages 143–155, 1989.

Jordi Luque, Guillem Cortès, Carlos Segura, Joan Fabregat, Javier Esteban, and Alexandre Maravilla. End-to-end photoplethysmography (PPG) based biometric authentication by using convolutional neural networks. In *2018 26th European Signal Processing Conference (EUSIPCO 2018)*, Roma, Italy, September 2018.

Microchip. 2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface. -https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf, 2008. [Online; accessed 10-June-2018].

J. Murphy and Y. Gitman. PulseSensor Open Hardware . -http://pulsesensor.com/, 2018. [Online; accessed 10-June-2018].

Obi Ogbanufe and Dan J. Kim. Comparing fingerprint-based biometrics authentication versus traditional authentication methods for e-payment. *Decision Support Systems*, 106:1–14, 2018. ISSN 01679236.

S. Patel, J. Pingel. Introduction to Deep Learning: What Are Convolutional Neural Networks? -https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html, March 2017. [Online; accessed 10-June-2018].

C. Segura et al. Automatic speech feature learning for continuous prediction of customer satisfaction in contact center phone calls. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 255–265, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49169-1.

Petros Spachos, Jiexin Gao, and Dimitrios Hatzinakos. Feasibility study of photoplethysmographic signals for biometric identification. *17th DSP 2011 International Conference on Digital Signal Processing, Proceedings*, pages 7–11, 2011. ISSN Pending.

V. Sze, Y. Chen, T. Yang, and J. S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

T. Fox-Brewster. Apple Face ID 'Fooled Again' – This Time By $200 Evil Twin Mask. -https://www.forbes.com/sites/thomasbrewster/2017/11/27/apple-face-id-artificial-intelligence-twin-mask-attacks-iphone-x/#255f55bf2775, November 2017. [Online; accessed 10-June-2018].

Telefónica I+D. Telefónica Research and Development. -https://www.tid.es/, 2018. [Online; accessed 10-June-2018].

The Apache Software Foundation. Apache License Version 2.0. -http://www.apache.org/licenses/LICENSE-2.0.txt, 2018.

Eben Upton. Raspberry Open Hardware. -https://www.raspberrypi.org/, 2018. [Online; accessed 10-June-2018].

Walber. Precision and Recall. -https://commons.wikimedia.org/wiki/File:Precisionrecall.svg, November 2014. [Online; accessed 10-June-2018].

Z. Zhang, Z. Pi, and B. Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, Feb 2015. ISSN 0018-9294. doi: -10.1109/TBME.2014.2359372.

Elif Derya Übeyli, Dean Cvetkovic, and Irena Cosic. Analysis of human ppg, ecg and eeg signals by eigenvector methods. *Digital Signal Processing*, 20(3):956 – 963, 2010. ISSN 1051-2004.

# Appendix A

# EUSIPCO 2018 Paper

# END-TO-END PHOTOPLETHYSMOGRAPHY (PPG) BASED BIOMETRIC AUTHENTICATION BY USING CONVOLUTIONAL NEURAL NETWORKS

*Jordi Luque*[1], *Guillem Cortès*[1], *Carlos Segura* [1],
*Alexandre Maravilla*[2], *Javier Esteban*[2], *Joan Fabregat*[2]

[1] Telefónica Research Edificio Telefónica-Diagonal 00, Barcelona, Spain
[2] Telefónica de España, Spain

## ABSTRACT

Whilst research efforts have traditionally focused on Electrocardiographic (ECG) signals and handcrafted features as potential biometric traits, few works have explored systems based on the raw photoplethysmogram (PPG) signal. This work proposes an end-to-end architecture to offer biometric authentication using PPG biosensors through Convolutional Networks. We provide an evaluation of the performance of our approach in two different databases: Troika and PulseID, the latter a publicly available database specifically collected by the authors for such a purpose. Our verification approach through convolutional network based models and using raw PPG signals appears to be viable in current monitoring procedures within e-health and fitness environments showing a remarkable potential as a biometry. The approach tested on a verification fashion, on trials lasting one second, achieved an AUC of $78.2\%$ and $83.2\%$, averaged among target subjects, on PulseID and Troika datasets respectively. Our experimental results on previous small datasets support the usefulness of PPG extracted biomarkers as viable traits for multi-biometric or standalone biometrics. Furthermore, the approach results in a low input throughput and complexity that allows for a continuous authentication in real-world scenarios. Nevertheless, the reported experiments also suggest that further research is necessary to account for and understand sources of variability found in some subjects.

***Index Terms***— photoplethysmogram signal, ppg, biometric authentication, biometric verification, convolutional neural networks

## 1. INTRODUCTION

User authentication based on monitoring the heart signal has raised the interest of the research community due to the increasingly popularity of wereable biosensors. Wrist-type photoplethysmographic (PPG) sensors have become a standard in health care and fitness applications owing to their capabilities for low cost and long term screening. Despite the fact that PPG signals can be easily obtained from the finger or by wrist-type wearables and smart-watches, it arises several questions about its potential and viability as a biometric trait, e.g. due to motion artifacts, as well as around the selection of appropriate biomarkers.

The PPG sensor is a non-invasive electro-optical method [1] that provides the PPG signal as illuminance variations measured by a photo-detector. Usually, a source of light is placed on a finger and a photo-detector placed right across the source detects the transmitted light reflected back. Shortly after the systole, the amount of blood in the arteries increase, thus reflecting it on the intensity of received light which increases too. The contrary occurs during the diastole, where the amount of blood in the arteries decreases leading to a decrease in the light observed by the photo-detector. Blood flowing characteristics are unique identifiers specific to different persons while they are similar enough to recognize the same person [2, 3, 4], keeping a strong relationship with person's anatomy and physiology as with the heart size and its dynamics.

Most of the approaches in the literature for biometric pulse identification rely both on involving Electrocardiography (ECG), based on the electrical activity of the heart, and on a carefully design, segmentation and extraction of expert features from the pulse signal [5, 6]. A decoupled approach which comprises mainly two stages is usually described [2, 7, 8]. Firstly, biomarkers or features are extracted from the pulse ECG or PPG signals, also known as front-end processing. Then, template feature vectors feed a second stage that performs model learning. Nonetheless, such features are designed by hand and strongly depend on a high expertise both on the knowledge of the addressed task and on acquisition nature of the pulse signal itself. For instance, in [2] an experiment on a group of 17 subjects was performed, where the authors studied four time domain characteristics, as time intervals, peaks and slopes from the PPG signals reporting successful accuracy rates of $94\%$ for human verification. In the work of [7], feature extraction on the PPG, ECG, EEG signals was performed based on eigenvector methods. Spachos et al. [3] studied four feature parameters, peak number, time interval, upward slope and downward slope. The study from [4] is intended for exploring the time domain features acquired from its first and second derivatives, where a group of 40 features were extracted and ranked based on a $k$-nearest neighbor algorithm. The authors in [8] perform a comparison of three methods based and proposed the pulsatile beat-by-beat correlation analysis, the rejection or acceptance of subject is performed based on the maximum similarity. Finally, more recent works [9] make use of Deep Belief Networks and Restricted Boltzman Machines as classifiers. With the advent of Deep Neural Network architectures, such as convolutional based neurons, end-to-end processing pipelines are gaining popularity by building architectures capable of learning features directly from raw data. For instance, in computer vision [10] or speech processing [11, 12] novel feature learning techniques are applied directly on the raw representations of images and audio, avoiding the signal parameterization or any other prior preprocessing.

This paper presents a new human verification approach using photoplethysmogram (PPG) signals and deep neural network modelling. The novelty of this work resides on the use of an end-to-end deep neural network architecture for both automatic extraction of biomarkers and low complexity allowing high continuous authentication rates. The proposed front-end, based on a Convolutional Neural Network, is jointly trained together with a dense neural net. Such architecture allows for a joint optimization of the extracted patterns

while maximizes the verification of the subject's identity. In addition and for development and evaluation purposes a new database, named as *PulseID*, is collected within a regular office environment, comprising 43 subject's IDs and their PPGs signals. Our proposed verification approach through neural network learning and classification appears to be viable as reported by the experiments performed on the Troika [13] and *PulseID* datasets. The results are encouraging, reaching AUCs around 83.1% by trials lasting just 1 second, both showing the potential of learned PPG biomarkers as a stand alone biometry and allowing for a continuous authentication in real-world scenarios.

## 2. PPG BASED VERIFICATION METHODOLOGY

### 2.1. Datasets

Two different datasets are employed in this work to conduct person verification experiments through PPG signals. Firstly, a new corpus was collected aiming to fulfil a need of, to the best of authors' knowledge, a public domain dataset specifically created for PPG biometric identification. For such a purpose, the authors collected a new PPG dataset, named as *PulseID*[1], in a quiet office environment. Secondly, aiming to verify the robustness of our approach in more challenging conditions, the Troika [13] dataset is used. In contrast, Troika recordings are acquired for subjects on a treadmill, walking and running at different speeds.

For the PulseID data acquisition, the pulse sensor described in [14] is employed. It is essentially a photoplethysmograph, a well known medical device used for non-invasive heart rate monitoring, consisting of a green LED and a photo-detector. The heart pulse signal that comes out of the pulse sensor is an analog fluctuation in voltage, with associated waveform known as photoplethysmogram or PPG, see figures 1(a) and (b). The pulse sensor responds to relative changes in illuminance. For a sensor placed in the subject's skin, the reflected light back to the photo-detector changes during each pulse due to blood flowing what is perceived as variations in the voltage signal. A Raspberry Pi 3 board was employed for hardware acquisition together with a popular analog to digital converter (ADC) MCP3008, accounting for 10 bits. The process of data acquisition was provided by 43 volunteers (31 male and 12 female) with ages ranging from 22 to 55. Subjects were seated down in a calmed, relaxed and quiet office environment while the recordings. The PPG sensor was attached to the fingertip of the right index finger by a belt. PPG acquisitions, lasting roughly one minute, were recorded from each subject and repeated 5 times along the same session. PPG analog signal was sampled at 200 Hz rate. For doing so, a python code was developed aiming to perform sampling synchronization while reading from the ADC and ensuring a tolerant averaged sampling rate deviation of $\mu = 13.32$ $\mu$s and $\sigma = 202.58$ $\mu$s per subject, see plot (c) in the figure 1 for an example. For comparison purposes, an ECG waveform from Troika is depicted in fig. 1(d). It is worth noting that no pre-processing is performed to the raw PPG acquired signals. It can be seen in the higher noise levels present in the acquired PPGs, where various artifacts are expected to be found: like analog circuit noises or medium illuminance changes, respiration or base deviation arising from movement.

[1]The *PulseID* dataset is available upon request from the authors and agreement of EULA for research purposes.

**Table 1**: Summary stats for both databases, Troika and PulseID. The "duration" column stands for the average duration in seconds of the total acquired samples per subject

| Dataset | Subjects | Gender (m/f) | Duration |
|---|---|---|---|
| PulseID | 43 | 31/12 | 240s. |
| Troika | 20 | 20/— | 317s. |

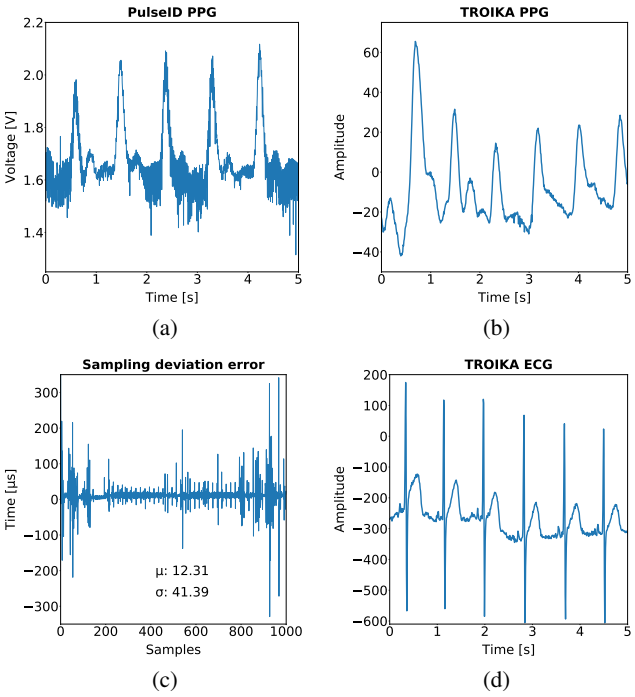



**Fig. 1**: Five seconds PPG excerpts from PulseID (a) and Troika (b) databases. (c) Time sampling deviation in $\mu$ seconds from the same PPG signal in (a). In (d), a five seconds ECG excerpt from Troika

In addition to PulseID database, the Troika dataset is employed to verify the robustness of our approach. Biometric identification using PPG should be possible even when the subject is in heavy physical motion. Therefore, Troika introduces a suitable database to benchmark learning models in practical day to day situations, by presenting higher heart signal variability and physical motion artifacts, is in theory a more challenging scenario compared to a relatively quiet office environment. During Troika recordings, subjects walked or ran on a treadmill at different speeds. The data was collected from 20 male subjects with ages ranging from 18 to 35. For each subject, the PPG signals were recorded from wrist by two pulse oximeters but only the first PPG channel is used in this work. The pulse signals were sampled at 125 Hz, see [13] for further details.

### 2.2. End-to-end biomarker learning

Convolutional Neural Networks (CNN) have become broadly applied reporting great success for instance in image recognition tasks [15, 16]. In the same sense, our deep CNN-based feature learning architecture makes use of local filtering and feature pooling, used at the output of the convolutional layers. The CNN architecture that we used as a basis for all our experiments is depicted in figure 2.

The end-to-end is validated and tested using 31 target and 12 impostor subjects with total of 43 subjects for the *PulseID* database. For the case of Troika, 15 target and 5 impostors accounting for a total of 20 subjects were used. The total number of enrolled subjects is 35 and 15, respectively. In the training phase, the waveform is homogeneously segmented in chunks of duration 1 second for each

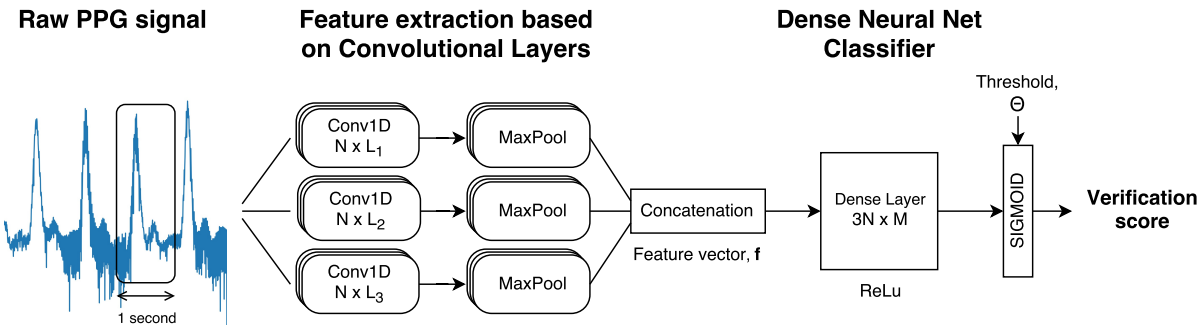



**Fig. 2**: Proposed Convolutional Neural Network architecture for end-to-end user verification using raw PPG signals. First, the raw signal is filtered by three parallel 1-D convolutional layers composed of $N$ filters of lengths $L_{1,2,3}$ followed by a global max-pooling operation. The resulting $3N$ features are then concatenated into the feature vector $\mathbf{f}$, which is used to perform the classification using a dense layer of dimensions $3NxM$ and the final layer of 1 output. ReLU activation function is used across all layers but the output layer, where a sigmoid activation is used to predict the verification score.

of the subjects. The table 2 reports on the partition set in terms of target and impostor trials for the case of excerpts lasting 1 second. Note that in a 60 bpm (beats per minute) Heart Rate there is one beat each second, there is no data-partition preprocess to ensure that a full peak is taken because every chunk contains a different signal, unless the subject Heart Rate is stable at 60bmp during all the acquisition. A combination of pairs (target,impostor) from the *Train* and *Validation* sets are used for network training and validation, performing parameter updating based on cross-entropy loss computed on the *Validation* set. The *Develop* set is used for final network testing and threshold selection but note that impostor trials are drawn from the same pool of identities than from previous sets. Finally, the *Test* set composed of unseen impostor subjects is employed for the fair assessment of the end-to-end proposed approach. Such data partitioning aims to prevent biasing and resembles a real use-case scenario in which cross-validated model is benchmarked against new enrolled users.

Note that the CNN-maxpooling feature learning architecture applies 1-dimensional convolutions and pooling operations performed along the time axis as previous works in [11]. Let's assume the PPG signal input to the CNN is a vector, $\boldsymbol{x}$, whose elements are raw PPG samples $\boldsymbol{x} = [x_k, x_{k+1} \dots x_{k+K}]$ where $x_k$ is the PPG sample shifted by a stride. In this work we used a value of 1 for time shifting and the $x_k$ sample with a fixed size from 1 second at 200 samples per second. The activations at the first convolutional layer comprise $N = 6$ filters and we denoted them as $\boldsymbol{h}_n = [h_1 \quad h_2 \cdots h_N]$. Therefore, the convolutional layer operation can be seen as a convolutional operation of each filter on the input raw PPG,

$$h_n = \theta\left(\boldsymbol{w}_n \boldsymbol{x}^T + b_n\right),$$

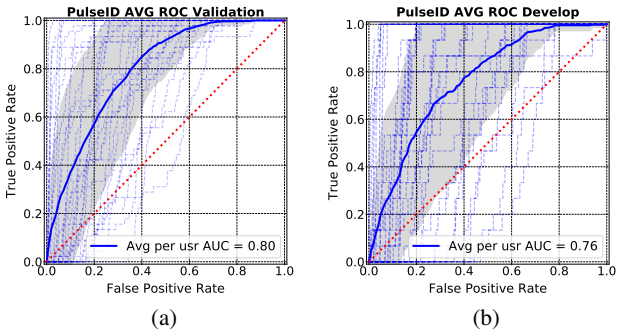
where $\theta(x)$ is the activation function corresponding to Rectified Linear Units (ReLU), $\boldsymbol{w}$ is the weighting vector and $b_n$ the bias term for filter $h_n$. Following the convolutional filters, max-pooling layers perform local temporal max operations over the input sequence, selecting the maximum in a window of $d$ size. More formally, the transformation at starting sample vector $n$, $\boldsymbol{c}_k^n$, corresponding to the filter output sequence of the first convolutional layer and $j$th filter is:

$$\max_{k-\frac{(d-1)}{2} \leq s \leq k+\frac{(d-1)}{2}} c_s^n$$

The pooling operation compacts even more the original signal by computing some stats, commonly such as maximum, mean and variance, from the CNN output. For this work maximum pooling is used by selection of the maximum values from the CNN filter outputs. Next, a flattering operation is performed, see figure 2, that aims at stacking together all the CNNs outputs, creating a feature vector ready to be presented to the network classifier. In overall, the end-to-end architecture comprises a total of 4 layers. In the input, a convolutional layer with different amounts of filters and lengths (see figure 2) followed by a max pooling layer. At the back-end, a fully connected neural net composed of 2 layers with 256 units.

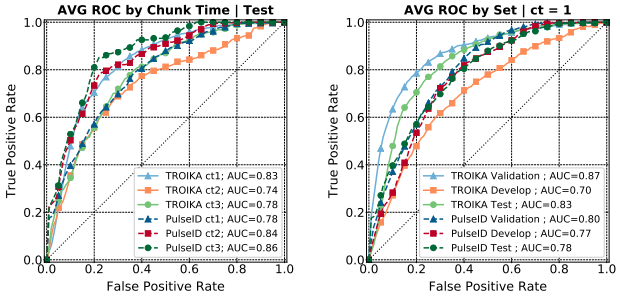
**Table 2**: Partition data for the different sets. Trials are expressed in seconds of signal and averaged per subject Since the trial size of the experiments showed is one second, the number of *Target* and *Impostor* data corresponds to number of trials or seconds

| Dataset | Label | Train | Validation | Develop | Test |
|---|---|---|---|---|---|
| PulseID | Target | 135 | 45 | 30 | 30 |
| | Impostor | 5,220 | 1,740 | 1,890 | 2,880 |
| | #Subjects | | 31 | | 12 |
| Troika | Target | 144 | 80 | 48 | 48 |
| | Impostor | 2,014 | 1,119 | 1,343 | 1,545 |
| | #Subjects | | 15 | | 5 |



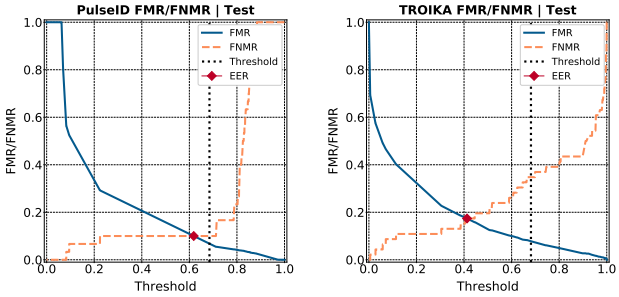

(a) (b)

**Fig. 3**: Average ROC for validation (a) and develop (b) sets using $N = 6$ filters for each filter sizes of $L_{1,2,3} = 50, 30, 20$. The painted area corresponds to the area within the standard deviation of the AUC. Dashed lines stands for each subject AUC curves




(a) (b)

**Fig. 4**: Average ROC comparison according to (a) the size of the chunk time, ct=1, 2, 3 s. and (b) set of data. The curves have been computed as in figure 3, averaged for all the subjects




(a) (b)

**Fig. 5**: False Match Rate (FMR) and False Non-Match Rate (FNMR) ratio plots as function of threshold $\Theta$. Threshold line $\Theta$, see fig. 2, corresponds to the operating point where the FMR is below 0.1 and the FNMR is minimum

ReLu units are employed in all layers, including CNNs, except for the output layer of the dense network, where we used a sigmoidal unit. The dense layer is employed as a back-end for the modeling of the salient features computed by previous convolutional steps and a Sigmoid activaction function is used in the output layer. It is worth to note that no dropout is used during network training.

The framework [17] has been developed in Keras [18] and using Tensorflow [19] as back-end. We do not perform an exhaustive search of network parameters and we restrict experiments by using few learned biomarkers. For instance, we compute 15 features before the dense layer for the reported 1s. experiments, see vector $\mathbf{f}$ in fig. 2. The network is trained using Stochastic Gradient Descent (SGD) attending to binary cross-entropy as a loss function and accuracy as a metric, with mini batches of size 270 composed of 135 target trials and 135 impostor ones. Given a *Train* set of PPG excerpts from a subject, at each training mini batch, the impostor samples are randomly picked up from the available pool of impostor chunks. Thus, in each training iteration, new impostor data is seen as an intent to maximize variability, see table 2. An early stopping criteria is also defined in order to speed up the training, yielding in most of the cases to few tens of training mini batches before reaching *patience* steps. The figure 3 shows the ROC curves, per *Validation* and *Develop* sets in PulseID and averaged per subjects, solid line, and its standard deviation, shadowed area. For the sake of comparison, the same curves are depicted in figure 4(b) per each dataset and partition.

## 3. EXPERIMENTAL RESULTS AND DISCUSSION

Although an exhaustive search of the best network architectures or a fully tuning of parameters is not performed, we experiment with different window and filter sizes. For the reported figures, we select $1, 2, 3$ second excerpts extracted from original raw PPG, homogeneously segmented and with no overlap for testing trials. The experiments are performed in PulseID data and best values, in terms of number of filters and size, are directly applied in Troika. Homogeneous segmentation of the input PPG likely degrades system performance due to few samples are taken into account for training, see table 2. However, it could be easily bypassed, e.g., by a randomly picking of excerpts thus increasing samples and segmentation variability in train and test. The figure 4(a) and table 3 report on the system performance for different trial sizes, ranging from 1 to 3 seconds. We can observe the generalization of validation results both in *Develop* and *Test* sets, showing high AUC values even for 1s. trial condition, AUC=0.78 and 0.83 per each dataset. Note the higher AUC degradation in Troika compared with the PPG data captured in the office condition and the AUC trend observed by increasing the $ct$ time, not observed in Troika likely due to motion artifacts. In overall, the results support the suitability of the end-to-end architecture in both datasets, although as observed in ROC curves fig.3, some subject's AUC present a not satisfactory behaviour suggesting more experimentation to understand possibles sources of such variability.

Another parameter to determine in authentication systems is the operating point or decision threshold. It controls the trade-off between usability, minimum FNMR, and security, minimum FMR. The fig. 5 reports on the Equal Error Rate (EER) 0.1 (a) and 0.174 (b) for a specific user in both datasets. Related to the algorithm complexity for trial decision, taking into account $L_i$ size of filters, pooling and dense layers operations, and 1s. trial lead to a number of MAC operations [20] around $26K$, that translates into roughly 20ms for a Raspberry in order to perform person authentication every second.

**Table 3**: Average AUCs for all subjects within the same experiment: $N = 6$ filters for each filter sizes of $L_{1,2,3} = 50, 30, 20$. the $\pm$variation corresponds to the AUC's standard deviation

| Dataset | Trial size | Validation | Develop | Test |
|---|---|---|---|---|
| PulseID | 1s. | 0.80±0.16 | 0.77±0.19 | 0.78±0.20 |
| | 2s. | 0.81±0.16 | 0.76±0.22 | 0.84±0.19 |
| | 3s. | 0.84±0.15 | 0.78±0.20 | **0.86±0.17** |
| Troika | 1s. | 0.87±0.09 | 0.70±0.16 | **0.83±0.12** |
| | 2s. | 0.73±0.30 | 0.66±0.21 | 0.74±0.24 |
| | 3s. | 0.85±0.14 | 0.71±0.16 | 0.78±0.18 |

## 4. CONCLUSIONS

An end-to-end architecture based on CNN is proposed to offer biometric authentication using learned biomarker directly from PPG raw signals. We reported evaluation results of the performance of our approach in two different datasets, Troika and PulseID. Our end-to-end authentication approach and automatic learned biomarkers show a remarkable potential as authentication biometric method. Trial size dependent experiments, reported AUCs ranging $[78.2\%, 86.4\%]$ and $[73.8\%, 83.2\%]$, averaged among target subjects on PulseID and Troika datasets, respectively. Furthermore, the proposed system results in a low complexity that permits for continuous authentication in real-world scenarios.

## 5. REFERENCES

[1] A. V. Challoner and C. A. Ramsay, "A photoelectric plethysmograph for the measurement of cutaneous blood flow," *Phys Med Biol.*, vol. 19(3), pp. 317–28, 1974 May.

[2] Y. Y. Gu, Y. Zhang, and Y. T. Zhang, "A novel biometric approach in human verification by photoplethysmographic signals," in *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003.*, April 2003, pp. 13–14.

[3] P. Spachos, J. Gao, and D. Hatzinakos, "Feasibility study of photoplethysmographic signals for biometric identification," in *2011 17th International Conference on Digital Signal Processing (DSP)*, July 2011, pp. 1–5.

[4] A. R. Kavsaoğlu, K. Polat, and M. R. Bozkurt, "A novel feature ranking algorithm for biometric recognition with ppg signals," *Computers in Biology and Medicine*, vol. 49, no. Supplement C, pp. 1 – 14, 2014.

[5] S. A. Israel et al., "Ecg to identify individuals," *Pattern Recognition*, vol. 38, no. 1, pp. 133 – 142, 2005.

[6] H. P. da Silva, A. Fred, A. Lourenço, and A. K. Jain, "Finger ecg signal for user authentication: Usability and performance," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Sept 2013, pp. 1–8.

[7] Elif Derya Übeyli, Dean Cvetkovic, and Irena Cosic, "Analysis of human ppg, ecg and eeg signals by eigenvector methods," *Digital Signal Processing*, vol. 20, no. 3, pp. 956 – 963, 2010.

[8] T. Choudhary and M. S. Manikandan, "Robust photoplethysmographic (ppg) based biometric authentication for wireless body area networks and m-health applications," in *2016 Twenty Second National Conference on Communication (NCC)*, March 2016, pp. 1–6.

[9] V. Jindal, J. Birjandtalab, M. B. Pouyan, and M. Nourani, "An adaptive deep learning approach for ppg-based identification," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2016, pp. 6401–6404.

[10] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 8595–8598.

[11] C. Segura et al., "Automatic speech feature learning for continuous prediction of customer satisfaction in contact center phone calls," in *Advances in Speech and Language Technologies for Iberian Languages*, Cham, 2016, pp. 255–265, Springer International Publishing.

[12] Y. Gong and C. Poellabauer, "How do deep convolutional neural networks learn from raw audio waveforms?," 2018.

[13] Z. Zhang, Z. Pi, and B. Liu, "Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 522–531, Feb 2015.

[14] J. Murphy and Y. Gitman, " PulseSensor Open Hardware ," http://pulsesensor.com/, 2017, [Online; accessed 19-October-2017].

[15] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, 1995.

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

[17] G. Cortès, J. Luque, J. Fabregat, and J. Esteban, " PulseID Keras code for development and testing purposes," https://guillemcortes@bitbucket.org/guillemcortes/pulseid-eusipco, 2018.

[18] F. Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.

[19] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[20] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.


Universitat Politècnica de Catalunya

Barcelona, 2018