
UNIVERSIDADE
DE PERNAMBUCO

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Programa de Pós-Graduação Acadêmica em Engenharia de Computação

Victor Kléber Santos Leite Melo

# A Deep Learning Approach to Generate Offline Handwritten Signatures Based on Online Samples

Dissertação de Mestrado

Recife, agosto de 2017


Universidade de Pernambuco
Escola Politécnica de Pernambuco
Programa de Pós-Graduação Acadêmica em Engenharia de Computação

Victor Kléber Santos Leite Melo

# A Deep Learning Approach to Generate Offline Handwritten Signatures Based on Online Samples

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-Graduação acadêmico em ENGENHARIA DE COMPUTAÇÃO da Universidade de Pernambuco como requisito parcial para obtenção do título de Mestre em Engenharia de Computação.

Prof. Dr. Byron Leite Dantas Bezerra
Orientador
Prof. Dr. Giuseppe Pirlo
Coorientador

Recife, agosto de 2017

*à minha família.*

# Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

O epígrafe é uma pagina onde o autor pode colocar uma citação ou pensamento que, de alguma forma, influenciou em seu trabalho acadêmico.

# Abstract

One of the main challenges of off-line signature verification is the absence of large databases. A possible alternative to overcome this problem is the generation of synthetic signature databases. In this work, a novel method for the generation of synthetic off-line signatures based on dynamic information is presented. In contrast to the state-of-the-art, we propose an synthesis approach under the perspective of supervised training, in which our learning model is trained to perform the task of "on-line signature to off-line signature conversion". The proposed approach is based on a Deep Convolutional Neural Network trained to learn how on-line handwritten manuscripts of the IRONOFF dataset are transformed into the off-line domain. The main goal of the proposed method is to synthetically enlarge existing off-line signature datasets based on on-line signature samples towards an improvement on the recognition rates of off-line signature verification systems. For these purposes, a machine-oriented evaluation on the BiosecurID signature dataset is carried out. We show that the synthetic samples generated with our proposed method achieve a verification performance similar to the one offered by real signatures and promising improvements of the Equal Error Rate results in comparison with and the current state-of-the-art method.

# Resumo

Um dos principais desafios de sistemas de verificação de assinaturas *off-line* é a ausência de grandes conjunto de dados. Uma alternativa possível para superar esse probema é a geração de assinaturas sintéticas. Neste trabalho é proposto um método para a geração sintética de assinaturas off-line baseado em informações dinâmicas. Em contraste com o estado-da-arte, o método método de síntese proposto se baseia na perspectiva da aprendizagem supervisionada, o nossa máquina de aprendizagem é treinada para realizar a tarefa de "conversão de assinatura *on-line* para *off-line*". O método proposto é uma *Deep Convolutional Neural Network* treinada para aprender como textos manuscritos *on-line* da base IRONOFF são transformados para o domínio *off-line*. O objetivo principal do método proposto é o de aumentar sinteticamente bases de assinatura *off-line* baseando-se em amostras *on-line* em direção a uma melhora nas taxas de reconhecimento de sistemas de verificação de assinaturas *off-line*. Para isso, uma avaliação na base de assinaturas BiosecurID é realizada. Mostra-se que as amostras sintéticas geradas pelo método proposto obtém uma performance de verificação similar aos oferecidos por assinaturas reais e uma melhora promissora no *Equal Error Rate* em comparação com o método do estado-da-arte.

# Contents

# List of Figures

# List of Tables

# List of symbols and abbreviations

| | |
|---|---|
| AHSVS | Automatic Handwritten Signature Verification System |
| EER | Equal Error Rate |
| FAR | Equal Error Rate |
| FRR | Equal Error Rate |
| DET | Equal Error Rate |
| ROC | Equal Error Rate |
| SVM | Equal Error Rate |
| CNN | Equal Error Rate |
| AER | Equal Error Rate |
| MSE | Equal Error Rate |

CHAPTER 1

# Introduction

In the modern society, biometric technology is used in several security applications for personal authentication. The aim of such systems is to confirm the identity of a given subject based on physiological or behavioral traits. In the first case, recognition is based on biological characteristics such as fingerprint, palm print, iris, face. The latter relies on behavioral traits such as voice pattern and handwritten signature [1].

The Handwritten Signature biometry stands as one of the primary methods for identity authentication. One of the reasons for its widespread is the fact that signature acquisition is easy, non-invasive, and most individuals are familiar with its use in their daily life [2]. Due to its convenient nature, signatures can be employed as a sign of confirmation in a wide set of documents, namely, bank checks, identification documents and a variety of business certificates and contracts.

As a behavioral trait, signatures present a high intra-user variability and are susceptible to spoof attacks, which is the attempt to forge the signature of a legitimate subject [1]. Two types of impostors are considered: casual impostors (producing random forgeries) when no information about authentic writer signature is known, and real impostors (producing skilled forgeries) when some information of the signature is used [3].

If a signature on a document is forged, this document is also considered invalid. Thus, preventing frauds in the signature verification process has been a challenge for researchers around the world. However, manual signature-based authentication of a large set of documents is a time-consuming and labor-intensive task. Hence, several Automatic Handwritten Signature Verification Systems (AHSVS) have been proposed to support this task. These systems aim to automatically decide if a query signature is in fact of a particular person or not.

An AHSVS is essentially a pattern recognition system that receives a signature as input, extracts a feature set from the data and classifies the sample using a template database as the reference. These datasets contain signatures digitized either by using an optical scanner to obtain the signature directly from the paper or using an acquisition device such as digitizing tablets or electronic pens with digital ink. The two domains are identified as offline (static) and online (dynamic), respectively. In the online modality, data is stored during the writing process and consists of a temporal sequence of the two-dimensional coordinates (x, y) of consecutive points, whereas in the offline case, only a static representation of the completed writing is available as an image. Moreover, each representation has specific attributes not present in the other [4].

In order to improve the performance of signature verification systems, bigger databases are required. Since the acquisition and distribution of real signatures arise legal and privacy concerns, the use of realistic synthetic signatures could be regarded as a good alternative. As a consequence, over the last years, several works on both online [5, 6] and offline [7, 8] signature synthesis have been carried out. These synthetically generated signatures show a similar behavior to real ones, thus enabling to enlarge existing databases and offering new possibilities for offline recognition.

The increasing use of portable personal devices capable of capturing on-line signature signals (e.g, Tablet PCs, PDAs, mobile telephones, etc) is producing a growing demand of person authentication applications based on signature signals. But in spite of its advantages, there are cases in which on- line signature verification is not yet commonly used because signatures are collected off-line. This is the case of many gov- ernment/legal/financial transactions that are performed daily. Also, off-line signature examination is the common type of criminal casework for forensic experts worldwide [7]. Fur- thermore, systems that combine both on- and off-line infor- mation are of interest in new scenarios where signatures are collected on a paper attached to a digitizing tablet (e.g. point- of-sale terminals).

There may be a case where the type of signature verification system used for training differs from that used for testing purpose. Though the test sample is of a genuine person, it might not be possible to prove with either of these systems alone. Hence, development of an integrated version of offline and online signature verification systems would be useful, either or both the offline and the online signature templates of the person being registered are recorded and an identification number is generated for that person. During testing, the test sample recorded is matched against the information available for that identification number in the database [9].

Some efforts have been performed on the generation of synthetic static data taking into account dynamic features during the synthesis process [10]. Among others, this type of synthesis approach presents the following possible practical applications: (i) generation of synthetic static samples to be fused with the original online signatures in order to improve the performance in an online verification scenario; (ii) enlarge existing offline signature databases; (iii) development of systems capable of integrating both online and offline samples interchangeably, towards a

unified signature biometry [11].

Despite the advancements for this category of synthesis, the synthetic offline samples created with the state-of-the-art systems still struggle to improve the recognition rates when used to enlarge existing offline signature databases.

In this work, we propose an approach based on Fully Convolutional Neural Networks (FCN) trained in a supervised manner, to learn the representation of offline manuscripts using as the input the online representation, expecting that the network predicts the corresponding static sample. This type of approach is interesting for our problem since the neural network learns how the dynamic features, in particular, the pressure, of the online representation of the manuscript translates into the static domain (pen on the paper).

In contrast to the models proposed in the literature [7, 8, 10], the approach proposed in this work is designed under the perspective of supervised training. In which a learning model is trained to perform the task of "online to offline conversion" using as the training data a data set containing both the online and offline versions of manuscripts. We expect that the Deep Neural Network model can learn how online information is transformed into the offline manuscript. Moreover, we expect that the trained model synthesizes offline signatures with improved discriminative power (i.e., better recognition rates).

## 1.1 Problem statment

Encouraged by the motivations depicted previously the goal of this dissertation can be stated as follows:

*"The goal of this work is to design an approach to generate synthetic offline handwriting signatures based on online data, modeling this problem as a supervised machine learning task, through a Deep Convolutional Neural Network, in order to enlarge offline signature datasets to improve offline signature verification systems recognition rates."*

This statement is developed through the following actions: (i) Creation and training of a Deep Neural Network model able to translate dynamic handwritten information into an offline manuscript (ii) Generation of an offline synthetic dataset based on a publicly available online signature dataset (iii) Compare the proposed approach's performance with state-of-the-art methods and to evaluate the closeness of synthetic signatures with respect to real signatures.

To achieve point (iii), machine-oriented validations are carried out. A state-of-the-art automatic signature verifier is used to evaluate our synthetic samples versus real offline signatures from a publicly available dataset. It is expected that similar results would be obtained with both the real and our proposed method synthetic signatures. Moreover, we also evaluate the use of synthetic signatures in the genuine enrollment set, analyzing the differences on the offline verification system recognition rates.

## 1.2 Dissertation structure

From this introduction, the remainder of this work is organized as follows:

Chapter **??** gives an overview on the topic of handwritten signature verification. We introduce handwritten signatures as a biometry trait, characterize how an automatic handwritten signature verification system works and summarize previous studies performed in the context of synthetic offline signatures from online data. In Chapter 3, we give a context and describe the concept of Deep Learning. Chapter 4 describes our proposed method. Chapter 5 presents the experimental protocol and Chapter 6 the results. Finally, in Chapter 7 we present the conclusion of this study together with suggestions for future work.

CHAPTER 2

# Handwritten Signature Verification

In this chapter, we give a brief reference to some essential concepts related to Handwritten Signature Verification, including definitions of notation and terminology used in the following chapters. First, we give an introduction and a general overview of the handwritten signature biometry, afterwards we discuss how an Automatic Handwritten Signature Verification system works, and finally, we give a brief overview of the state-of-the-art on offline signature synthesis based on online data.

## 2.1 Handwritten Signature: a behavioral multimodal biometry

The term “Biometrics” is derived from the Greek word “bio-metriks”. In which “bio” means “life” and “metrics” means “to measure”. Biometrics refers to the measurements and statistical analysis of unchanging biological characteristics peculiar to an individual. Biometric systems are a constantly growing technology [1] and have been introduced as forms of identification and access control. Biometric identifiers are a unique measurable characteristic used to distinguish and describe individuals [12].

Biometric systems are often categorized as physiological or behavioral [13]. The physiological category is characterized by measurements of the body. Examples include fingerprint, palm veins, face recognition, DNA, palm print, hand geometry, iris/retina pattern, and body scent. On the other hand, behavioral biometrics are acquired traits by an individual and are related to the pattern of behavior of a person. They include typing rhythm, gait, temperament, voice, and handwritten signatures [14].

Most biometric identifiers require a special type of device for security and control of human identity. However, handwritten signature based biometric systems can be realized requiring no sensor except a pen and a piece of paper. According to [15] handwritten signatures can be considered the most legally and socially attributes accepted for person identification. Moreover, the challenge that comes with signature-based authentication is the need for high accuracy results to avoid false authorization or rejection.

Handwritten signature authentication is based on systems for signature verification. Whether a given signature belongs to a claimed person or not is decided through a signature verification system, which ultimately strives to learn the manner in which an individual makes use of their muscular memory (hands, fingers, and wrist) to reproduce a signature [16].

A generic handwritten signature based biometric system is shown in Figure 1. Once the user $\boldsymbol{Y}$ deposits the signature, a sensor digitalizes the sample. Later, a feature matrix $\boldsymbol{X}$ is built with the information extracted from the input sample. Next, the systems typically have two stages: enrollment $\boldsymbol{X_E}$ and recognition $\boldsymbol{X_R}$. The former builds a system database $\boldsymbol{D}$ where the users store their reference signatures as a set of templates, whereas the latter is used to recognize, identify or verify the identity of a user, who typically claim to be one of the registered users. Then, a score $\boldsymbol{S}$ is obtained according to the similarity of the questioned sample to the claimed template. Finally, the system accepts or rejects the questioned sample.

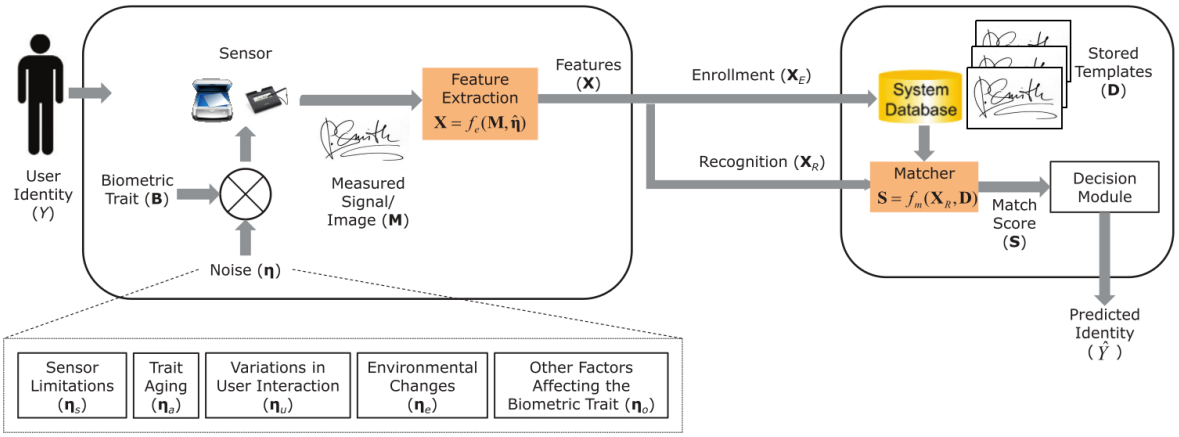



Figure 1 – Overview of a typical handwritten signature based system. Figure adapted from [14].

As Figure 1 shows, the signature acquisition sensor can be either an optical scanner or an acquisition device such as a digitizing tablet. These two different acquisition tools characterize the two classes of signatures, namely: static and dynamic.

In the static modality, also referred to as offline, an optical scanner is used to obtain the signature directly from the pen on the paper, and only the digital image of the signature is available, see Figure 2 (a). In the dynamic mode, also called online, signatures are acquired through a graphic tablet or a pen-sensitive computer display, see Figure 2 (b). In this mode, data is stored during the writing process and consists of a temporal sequence of the two-dimensional

coordinates $(x, y)$ of consecutive points.

(a) (b)

Figure 2 – Different signature acquisition methods. (a) a signature scanned from paper and (b) digitizing tablet Wacom STU-500 [17].

Specifically, the online modality does not convey information about the overall shape of the signature, the width of the strokes and the texture of the ink on the paper [10]. The offline representation, however, has lost all dynamic information about the manner in which the signature is signed during the acquisition process. As a result, features such as pen trajectory, which can be easily computed in the online domain, can only be inferred from a static image [18]. An example of a colored offline signature and a plotted matching online signature can be found in Figure 3.

Figure 3 – An offline and a matching online signature sample. Figure extracted from [19].

Once the signature sample is acquired, during the enrollment phase, the system tries to create the subject identity based on behavioral features in the signature. Because of the way we sign, however, it is a subtle task. The rapid movement behind the signature creation is determined by a motor program stored in the brain of each signer applied to tools such as the pen on the paper [20]. According to [21] there is a wide variety of human and social aspects that might affect the way we produce our handwritten signature, it might be influenced by country, age, time, habits, psychological or emotional state. The variability created on the signing process must be taken into account in the signature authentication process.

In fact, the unpredictable intra-personal variability, i.e. the similarity between signatures executed by the same writer, is a crucial challenge of signature-based biometric systems. This variability can be attributed to the several sources of noise ($\eta$) that distort the measured trait. According to Figure 1, the intra-personal variability affecting the measured sample $\boldsymbol{M}$ can be characterized by several variables. These variables include sensor limitations such as resolution or sample rate; biological aging effects or cognitive-motor impairments; user interaction with the sensor; environment changes like background noise and other factors as consequence of the individuals' mood, hurry or unwillingness to cooperate. The intra-personal variability effect is illustrated in Figure 4.

Figure 4 – Superimposed genuine signatures of the same writer. A high intra-personal variability can be noticed. Extracted from [22].

Another challenge faced by signature-based biometric systems is the unpredictable inter-personal variability, i.e. the similarity between signatures executed by different writers. In a signature-based system, inter-personal variability is mainly attributed to frauds related to malicious people faking the identity of signers. Figure 5 illustrates a visual comparison between genuine signatures and forgeries.

Figure 5 – The first column of signatures are genuine references, the following three samples are questioned signatures. How many forgeries would you be able to detect?[1] Signature images extracted from [23].

In the field of signature verification, forgeries are generally classified into two types.

- The first one is the random forgery which is created in a situation which an impostor who has no information about the person or the shape of the original signature tries to verify the identity of a signer by using his genuine signature. The random forgery test is a typical test used in access control and commercial transactions.
- The second type is the skilled forgery, represented by a proper imitation of the genuine signature model. The forger has access for both the user's name and signature, and learns the signature of a signer and tries to reproduce it with a similar intra-class variability. This test is the most relevant in signature verification for its impact in forensic applications in signature forgery detection.

## 2.2 Automatic Handwritten Signature Verification

An Automatic Handwritten Signature Verification System (AHSVS) is conceptually a pattern recognition application. Pattern recognition is one of the most important and active fields of research. During the past few decades, there has been a considerable growth of interest in problems of pattern recognition, and in the last few years, many methods have been developed in this area, in particular on handwriting recognition and signature verification [24].

As any Pattern Recognition system, an AHSVS has three phases: data acquisition and pre-processing, feature extraction and classification [2].

In the first step, the signatures are acquired and preprocessed, the main goal here is to convert them into a format suitable for the modeling process, correcting geometric distortions and removing noise related to the signature acquisition sensor. Afterwards features are extracted and stored in a knowledge database. On the classification step, the extracted features are used to distinguish between genuine and forged signatures. Therefore the Signature Verification task is, in essence, a two-class classification problem, in which the system's prediction to the input signature sample is either genuine or fraud.

Verification errors occurring in AHSVS are usually categorized as two types [25]. On the one hand, a genuine signer may be rejected by the system as a potential impostor (e.g. it could happen when the signer carelessly executes his/her signature), resulting in what is denoted a Type-1 error or False Rejection. On the other hand, a skilled forger might be able to produce a sample which would be accepted as genuine, resulting in what is called a Type-2 error or False Acceptance.

In order to improve the performance of signature verification systems, bigger databases are required. The amount of data available for each user is often insufficient in real applications.

[1] From left to right, top to bottom (F means Forgery and G means Genuine): FGF FFG GFF

During the enrollment phase, users are often required to supply only a few samples of their signatures. In other words, even if there is a significant number of users enrolled in the system, a classifier needs to perform well for a new user, for whom only a small set of samples are available. Since the acquisition and distribution of real signatures arise legal and privacy concerns, the use of realistic synthetic signatures could be regarded as a good alternative.

## 2.3 Off-line Signature Synthesis Using On-line Samples

One could synthesize an offline signature 2D image by interpolating the tracing, using a spline function for instance, between the digitized online points in addition to some morphological operators to enlarge the tracing to the desired stroke width. However, grayscale information would still be missing.

TODO - quero colocar aqui uma visão geral das propostas existentes do estado da arte

assessment [26]

rabasse 2008 [27]

ferrer synthetic online to synthetic offline [8]

[10]

[28]

CHAPTER 3

# Neural Networks and Deep Learning

This chapter introduces and gives a brief overview of the theoretical foundations of Deep Neural Networks. In the first section, basic concepts of Artificial Neural Networks are presented, starting from the original models. The next section gives an introduction and a general overview of the deep learning research field, discussing the recent advancements that enabled training Deep Neural Networks successfully. Finally, we briefly describe the fundamentals of the Convolutional Neural Network and present the Fully Convolutional Networks, which is a model used in this work and we present the techniques used for training our model.

## 3.1 Artificial Neural Networks

A biological neural network is an essential part of human brain. The human brain is a highly complex information processing system capable of interpreting large amounts of information and making decisions. It is a complex, non-linear and parallel "computer" consisting of millions of connected neurons [29]. In many tasks, the human brain is more efficient than computers. For instance, the human brain can recognize a familiar face in about 100-200 ms, while modern computers require minutes or even hours for solving the same problem [29].

Based on examples and feedback from the "teacher", our brain allows us learning how to distinguish an apple from an orange or recognize letters. Moreover, even without the "teacher", we are still able to group similar patterns. Those and other strengths of human brain challenged scientists to emulate those processes by researching how to use machines for tasks that are common for humans. Moreover, one of the concepts that appeared as the result of that research is the Artificial neural network (ANN) concept.

### 3.1.1 Artificial Neuron

The first model to simulate a single neuron, which is the elemental building block of neural networks, was the perceptron [30]. A single neuron implements a mathematical function given its inputs, to provide an output, as described in Equation 3.1 and illustrated in Figure 6.

$$y = f(\sum_{i=1}^{n} x_i w_i + b) \tag{3.1}$$

In this equation, $x_i$ is the input $i$, $w_i$ is the weight associated with input $i$, $b$ is a bias term and $f$ the activation function.

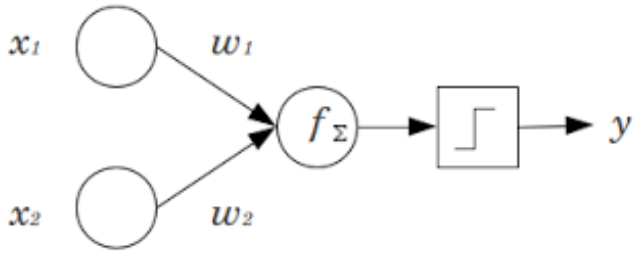



Figure 6 – Perceptron representation. $x_1$ and $x_2$ represent the input signal, $w_1$ and $w_2$ the weights, $f_\Sigma$ is the activation function (in this case, a step function) and the output signal is given by $y$.

Models based perceptrons, have severe limitations. An evaluation by Minsky and Papert [31] showed that a perceptron cannot model data that is not linearly separable, such as a simple XOR operator. It was observed that "for data sets that are not linearly separable, the perceptron learning algorithm will never converge" [32]. This observation is related to the perceptron's limited representational power, the learning rule only converges to the correct solution if the data is linearly separable. The Multilayer Perceptron (MLP) with a single hidden layer, however, has been proven [33] to approximate any continuous function on a compact input domain to arbitrary precision. For this reason MLPs are said to be *universal function approximators* [33].

## 3.2 Multi-layer Neural Networks

Multi-layer Neural Networks such as the MLP consist of stacking several neuron units together. Specifically in the MLP, neurons are stacked in one layer connecting these stacks sequentially, without connections between the neurons in the same layer, as it is shown in Figure 7. Each neuron in the MLP is normally fully connected with all the neurons in the next layer, with its own set of weights. The first layer of an MLP is usually the input sample and the last layer is the output of the network. The layers between the input and output layers are called hidden layers.

An MLP can be thought of as a function that maps from input to output vectors, parameterized by the neuron connection weights. The output of a layer is calculated by applying the

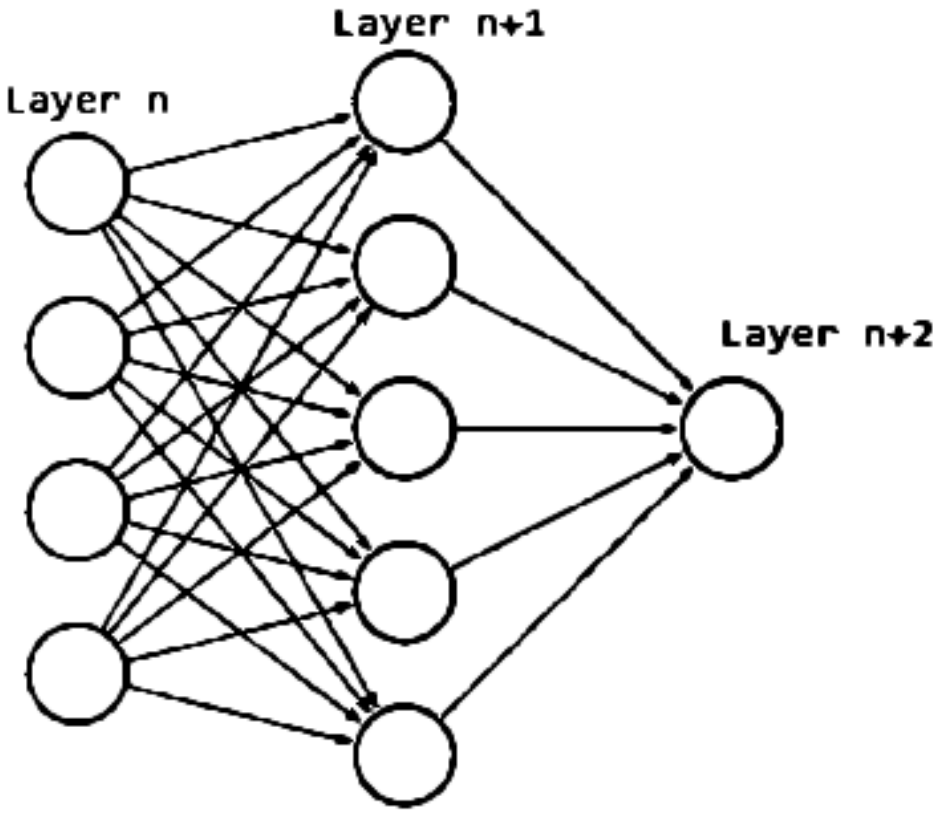



Figure 7 – Multilayer perceptron representation. Each layer contains several perceptron units, which are then connected to units in the subsequent layer.

neuron activation function for all neurons on the layer, as noted in Equation 3.2

$$y^{(l)} = f(W^{(l)} y^{(l-1)} + b^{(l)}) \tag{3.2}$$

where $W^{(l)}$ is a matrix of weights assigned to each pair of neurons from layer $l$ and $l-1$, and $b^{(l)}$ is a vector of bias terms for each neuron in layer $l$. Calculating the output starting from the first hidden layer, up to the output layer is also referred to as the *forward propagation* phase.

### 3.2.1 Loss Function

In order to train the model, an objective function, also called loss function, is defined. This function measures the compatibility between a prediction (e.g. the class scores in classification) and the ground truth label. The objective of the training is to minimize the sum (or, equivalently, the mean) of this error function applied to all examples in the dataset. Commonly used loss functions are the Mean Squared Error function (MSE), and the Cross Entropy (CE). Equations 3.3 and 3.4 describe the MSE and CE, respectively, for an sample of the dataset.

$$E = \frac{1}{N} \sum_{c}^{N} (t_{(c)} - y_{(c)})^2 \tag{3.3}$$

$$E = -\sum_{c}^{N} (t_{(c)}\, log\, y_{(c)})^2 \tag{3.4}$$

where $y_{(c)}$ is the output of unit $c$ in the last layer, $t_{(c)}$ the true label $t$ for unit $c$ and $N$ is the number of units on the last layer.

### 3.2.2 Backpropagation

The phase called backpropagation consists in minimizing the error $E$ between the network output and the expected target. The algorithm works by calculating the derivatives of the error function with respect to the model's parameters (weights and biases) and propagating the error from the output layers, back to the initial layers, one layer at a time, in order to update the weights of the neuron connections to minimize the error $E$. The weights are updated conforming Equation 3.5

$$w(t+1) = w(t) - \alpha\ \Delta\ E(w(t)) \tag{3.5}$$

where $E$ is error measure defined as loss sum over the entire training set, and $t$ indicates iterations (time steps). The $\Delta(E)$ is the gradient vector, which is computed by applying the chain rule on the layers of the NN [34]. The parameter $\alpha$ is a heuristic, called learning rate. The learning rate values help to avoid convergence to a local-minimum or maximum point. In order to secure the convergence of the backpropagation training algorithm and avoid oscillations in a steep direction of the error surface a small learning rate is chosen $(0 < \alpha < 1)$.

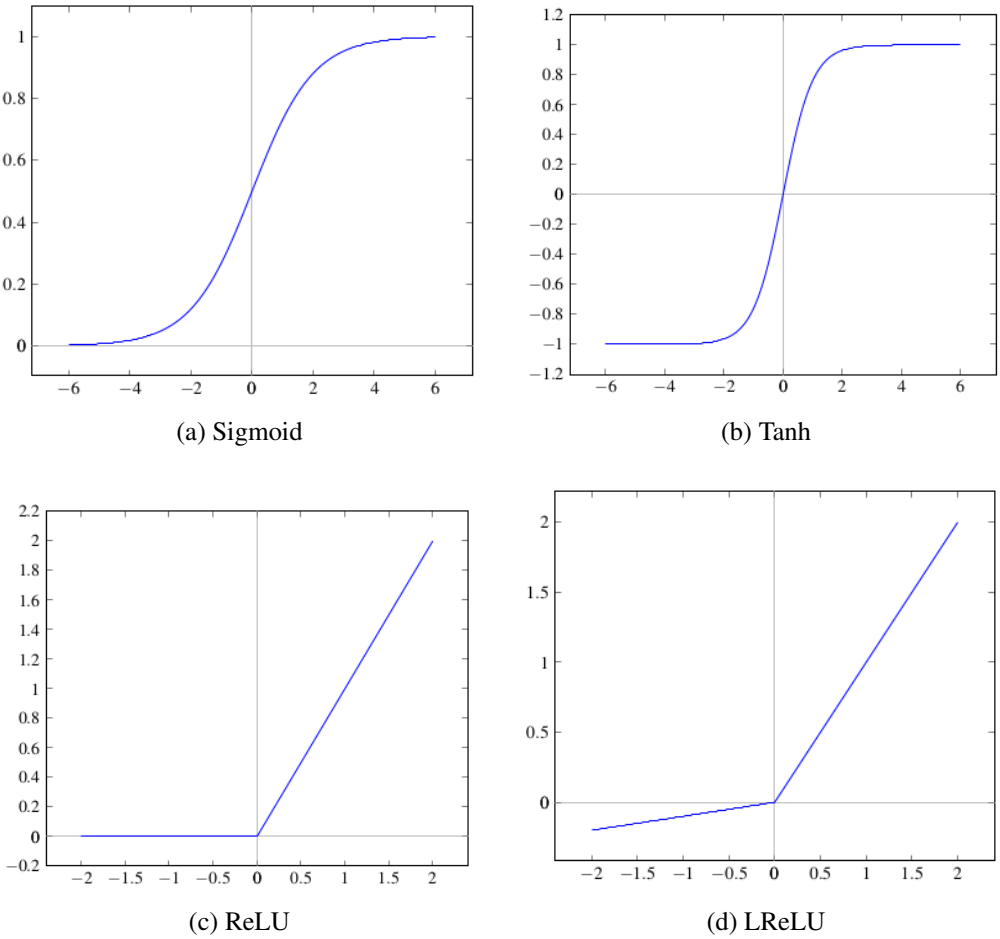
### 3.2.3 Activation function

#### 3.2.3.1 Sigmoid

The sigmoid activation function is a non-linear function in the range of $]0, 1[$ and has the mathematical form defined in Equation 3.6

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.6}$$

and is shown in the Figure 8 (a). It takes a real-valued number and transforms it to the range between 0 and 1. In particular, large negative numbers become 0 and large positive numbers become 1. The sigmoid function has seen frequent use historically since it has a natural interpretation as the firing rate of a neuron: from not firing at all (0) to fully-saturated firing at an assumed maximum frequency (1).




(a) Sigmoid (b) Tanh

(c) ReLU (d) LReLU

Figure 8 – Neural network activation functions.

#### 3.2.3.2 Tanh

The Hyperbolic Tangent (tanh) non-linearity is shown in Figure 8 (b) and is defined in Equation 3.7. It limits a real-valued number to the range of $[-1, 1]$.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.7}$$

#### 3.2.3.3 ReLU

The Rectified Linear Unit has become very popular in the last few years. It computes the function

$$f(x) = max\,(0, x) \tag{3.8}$$

in other words, the activation is simply thresholded at zero, see Figure 8 (c).

#### 3.2.3.4 LReLU

Leaky ReLUs are one attempt to fix the "dying ReLU" problem. Instead of the function being zero when x < 0, a leaky ReLU will instead have a small negative slope (of 0.01, or so).

That is, the function computes where $\alpha$ is a small constant. Some people report success with this form of activation function, but the results are not always consistent.

$$f(x) = \begin{cases} \alpha x, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases} \tag{3.9}$$

Figure 8 (d)

## 3.3 Deep Neural Networks

Deep architectures are characterized by the multiple levels of non-linear operations contained on a neural network. While many of the early successful applications of neural networks used shallow architectures (up to 3 hidden layers), the mammal brain is organized in a deep architectures. The brain appears to process information through multiple stages, which is particularly clear in the primate visual system [35].

It was observed in many experiments that deep networks are harder to train than shallow networks, and that training deep networks often get stuck in apparent local minima (or plateaus) when starting with random initialization of the network parameters. Deep Neural Networks have been investigated for decades, but training deep networks consistently yielded poor results.

The proposal of the Convolutional Neural Network [36] and some recent theoretical discoveries made training deep neural networks feasible. The CNN is a particular type of deep, feedforward neural network that is easier to train and generalized better than networks with full connectivity between adjacent layers [37]. Besides, the theoretical studies include unsupervised training of the layers [38], Rectifier Linear Unit (ReLU) [39] as the activation function and the regularization technique dropout [40].

## 3.4 Convolutional Neural Networks

CNNs combine three architectural ideas: local receptive fields, shared (tied) weights and spatial or temporal sub-sampling [41]. When trained with appropriate regularization, CNN can achieve superior performance on visual object recognition and image classification tasks [37]. The CNN is composed of several layers of trainable filters (convolutional layers) and local neighborhood pooling operations (pooling layers) stacked in an alternating sequence starting with the raw input images. To illustrate the concept, Figure 9 shows an example of architecture of a CNN, namely, the Lenet [41], one of the first applications of the CNN used for handwriting recognition.

Figure 10 provides an example of the filters learned on a convolutional layer of a network trained for object classification, showing that this type of architecture is capable of learning interesting feature detectors, similar to edge detectors.

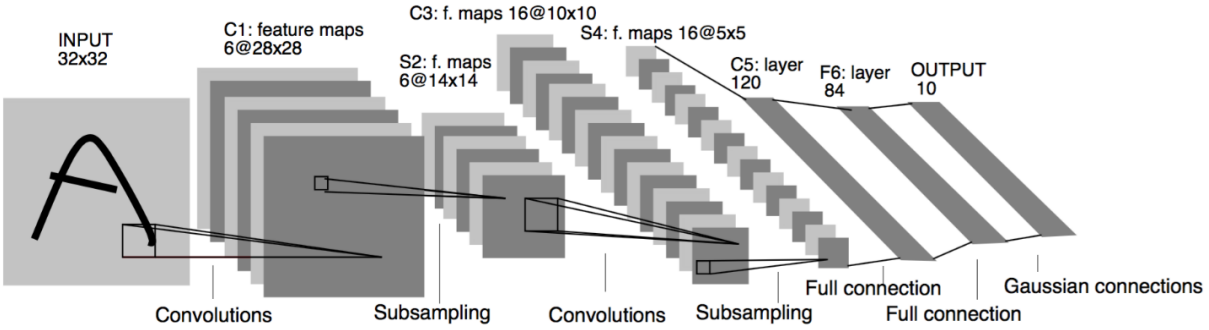



Figure 9 – Lenet [41], a convolutional neural network used for handwriting recognition.

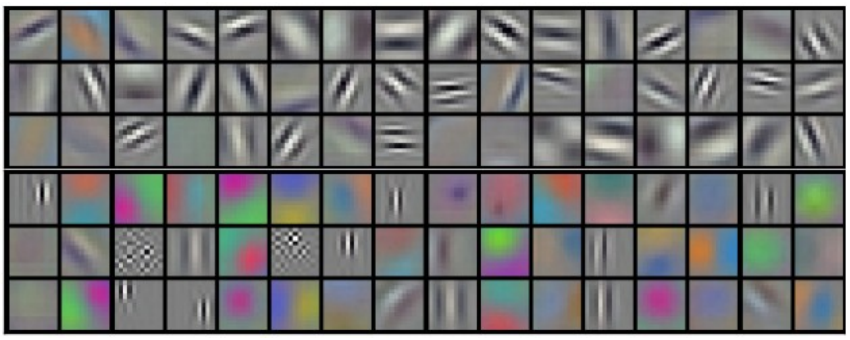


Figure 10 – Example of feature maps learned by the CNN proposed by [42] for object classification.

### 3.4.1 Convolutional Layer

Convolutional layers have trainable filters (also called feature maps) that are applied across the entire input [36]. For each filter, each neuron is connected only to a subset of the neurons in the previous layer. In the case of 2D input (such as images), the filters define a small area (e.g. 3x3 or 5x5 pixels), and each neuron is connected only to the nearby neurons (in this area) in the previous layer. The weights are shared across neurons, leading the filters to learn frequent patterns that occur in any part of the image.

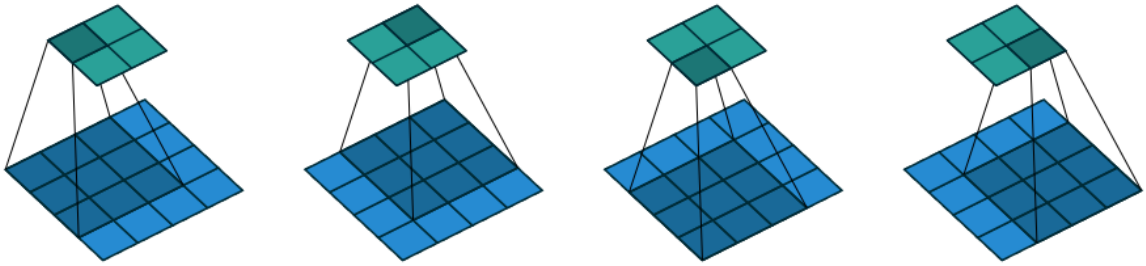
The definition for a 2D convolution layer is presented in Equation 3.10. A 2D convolution layer is the application of a discrete convolution of the inputs $y^{(l-1)}$ with a filter $w^{(l)}$, adding a bias $b_{(l)}$ , followed by the application of a activation function $f$:

$$y_{rc}^{(l)} = f(\sum_{i=1}^{N_r}\sum_{j=1}^{N_c} y_{(r+i-1)(c+j-1)}^{(l-1)} w_{ij}^{(l)} + b^{(l)}) \tag{3.10}$$

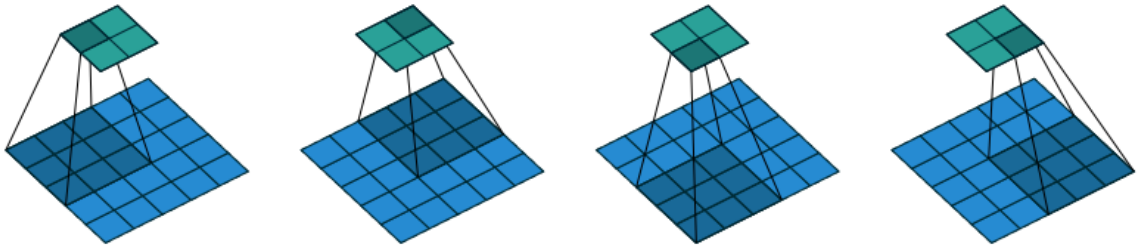
where $y_{rc}^{(l)}$ is the output at position $\{r, c\}$, $N_r$ and $N_c$ are the number of rows and columns, respectively, of the 2D filter, $w_{ij}^{(l)}$ is the filter value at position $\{i, j\}$, $y_{(r+i-1)(c+j-1)}^{(l-1)}$ is the $\{r+i-1, c+j-1\}$.

The equation above is defined for all possible applications of the filter, as in Figure 11 (a), that is, for $r \in \{1, ..., I_r - N_r + 1\}$ and $c \in \{1, ..., I_c - N_c + 1\}$, where $I_r$ and $I_c$ are the number of rows and columns in the input to this layer. The convolutional layer can either apply the filters

for all possible inputs [2], or use a different strategy. Instead of applying the filter for all possible $\{r, c\}$ pairs, only the pairs with distance $s$ are used, which is called the stride. A stride $s = 2$ [3] is equivalent to apply the convolution for half of the possible pairs, as in Figure 11 (b).



(a) Convolving a 3x3 kernel over a 4x4 input, with 1x1 stride.



(b) Convolving a 3x3 kernel over a 5x5 input, with 2x2 stride.

Figure 11 – Extracted from [43].

## 3.5 Fully Convolutional Network Architecture

The Fully Convolutional Network (FCN) [44] is a CNN modified for dense predictions. This architecture was developed under the observation that although a typical CNN is designed for spatial inputs, such as images, it normally discard spatial information in their fully connected layers (FC).

FCNs are, thus, rooted in the observation that spatial filters, which are learned during the CNN training, are useful for extracting low-level features, but FC layers are needed to incorporate high-level reasoning. The main idea of the FCN architecture is to convert the FC layers of a CNN into convolutional layers, expecting that the network retains the ability to learn high-level information and at the same time preserve spatial information. The FCN becomes, therefore, "fully convolutional" by having end-to-end convolutional layers. While CNNs are typically built in a sequence of convolutional, pooling, and fully connected layers, FCN adds an expanding path

[2] Animated representation can be found in: <https://raw.githubusercontent.com/vdumoulin/conv_arithmetic/master/gif/no_padding_no_strides.gif>

[3] Animated representation of strides can be found in: <https://raw.githubusercontent.com/vdumoulin/conv_arithmetic/master/gif/no_padding_strides.gif>

built with a transposed convolutional layer [44]. The expanding path recovers spatial information by merging features skipped from the various resolution levels on the contracting path.

Unlike a CNN, which learns a general, nonlinear function that characterizes the input, FCNs learn an end-to-end nonlinear mapping from one input image to another. Because pixel-wise prediction is used, the label space is also transformed from a scalar to a two-dimensional image, where each pixel value represents the object class of its corresponding input pixel. An FCN with an input image and label is illustrated in Figure 12

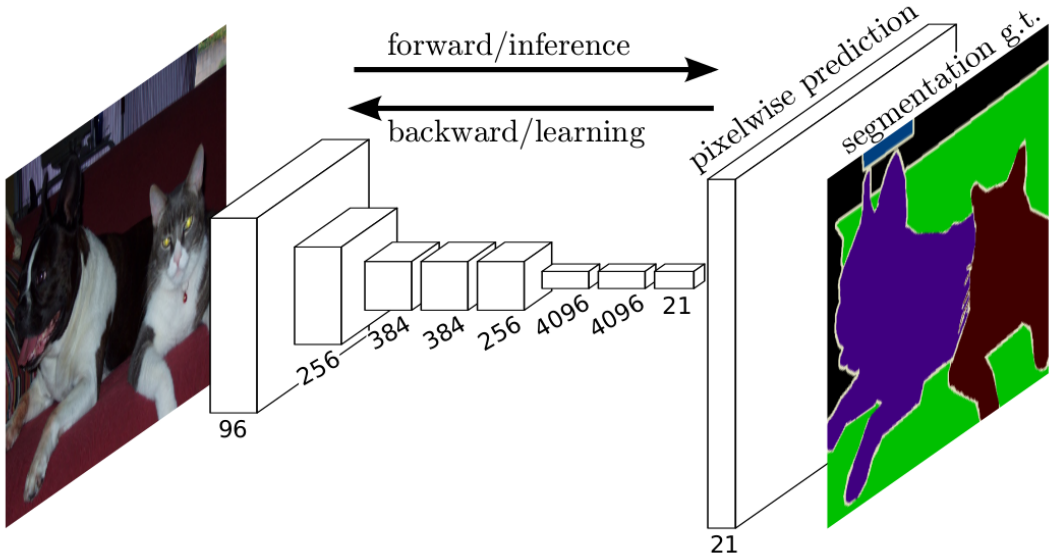



Figure 12 – Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks. Extracted from [44]

### 3.5.1 Transposed Convolution

Transposed convolutions - also called deconvolution or backwards convolution - work by swapping the forward and backward passes of a convolution. One way to put it is to note that the kernel defines a convolution, but whether it's a direct convolution or a transposed convolution is determined by how the forward and backward passes are computed.

The need for transposed convolutions generally arises from the desire to use a transformation going in the opposite direction of a normal convolution, i.e., from something that has the shape of the output of some convolution to something that has the shape of its input while maintaining a connectivity pattern that is compatible with said convolution. For instance, one might use such a transformation as the decoding layer of a FCN or to project feature maps to a higher-dimensional space. Figure 13 illustrates the general backwards convolution process [43].

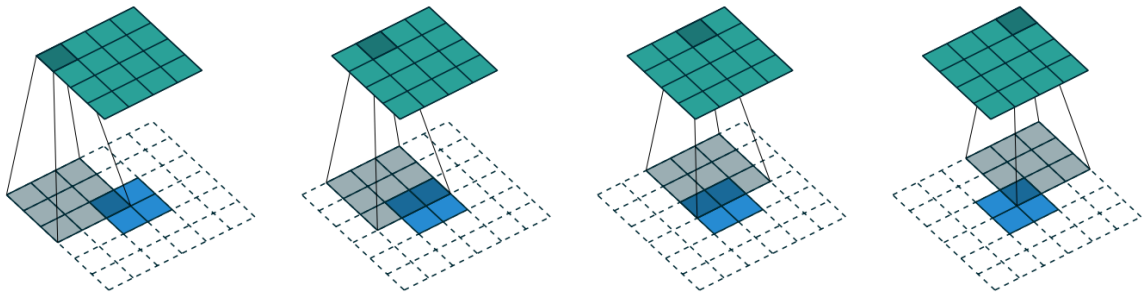


Figure 13 – Upsampling with transposed convolutions. Upsampling is done by padding (white) the original pixels (blue) and convolving them with a filter (gray). The result is an upsampled image (green). Here a stride of one is used, illustrated from left to right. Extracted from [43].

## 3.6 Neural Networks training

### 3.6.1 Weight initialization

When the training of the neural network starts, the initial values of the weights and biases need to be provided. Some techniques to weight initialization have been proposed and shown to be useful to smooth the model convergence.

#### 3.6.1.1 Uniform

The uniform initialization is the most simple. The model parameters are initialized through a random distribution in the range of $[l1, l2]$, where $l1$ and $l2$ are two constants.

#### 3.6.1.2 Glorot

This weight initialization technique was proposed in [45]. The dynamic of the activation functions and the gradients weights were studied and they found that the neural network converges faster using the weight initialization described in Equation 3.11.

$$W \sim U\left[\sqrt{\frac{6}{f_{in} + f_{out}}}\right] \tag{3.11}$$

where $W$ are the weights, $U$ is an uniform distribution, and $f_{in}$ and $f_{out}$ are the number of units in the previous layer and the next layer, respectively.

### 3.6.2 Optimization algorithm

Como dito na subseção 3.1.2, de posse dos gradientes dos parâmetros com relação à função de custo, calculados na fase Backward, é necessário atualizar os parâmetros de modo a

minimizar a função de custo da RNAs, com o objetivo de que a mesma aprenda a desempenhar a tarefa de maneira satisfatória.

#### 3.6.2.1 Stochastic Gradient Descent

Um dos algoritmos mais conhecidos para atualizar os parâmetros utilizando os gradientes dos mesmos com relação à função de custo é o Gradiente Descendente Estocástico (SGD, Stochastic Gradient Descent), a estocasticidade desse algoritmo ocorre devido a seleção de um exemplo aleatório do banco de dados para apresen- tado ao modelo. A Equação (20) mostra a aplicação da atualização SGD em um determinado peso de uma RNA. The SGD algorithm, as defined in [32], is described below at a high-level. The inputs are: W - the weights and biases of the network, (X, y) - the dataset, batch size - the number of training examples that are used in each iteration, and $\alpha$, the learning rate. For notation simplicity, all the model parameters are represented in W . In practice, each layer usually defines a 2-dimensional weight matrix and a 1-dimensional bias vector. In summary, SGD iterates over mini-batches of the dataset, performing forward-propagation followed by a back-propagation to calculate the error derivatives with respect to the parameters. The weights are updated using these derivatives and a new mini-batch is used. This procedure is repeated until a convergence criterion is reached. Common convergence criteria are: a maximum number of epochs (number of times that the whole training set was used); a desired value for the cost function is reached; or training until the cost function shows no improvement in a number of iterations.

#### 3.6.2.2 Adaptive Moment Estimation

A Estimação de Momento Adaptativa (ADAM, Adaptive Moment Estima- tion) é um método de otimização estocástica que necessita apenas de gradientes de primeira ordem e calcula taxas de aprendizagem adaptativas de maneira individual para os parâmetros a partir da estimação do primeiro e segundo momento dos gradientes. As Equações (24a) - (24e) mostram os cálculos necessários para utilizar este algoritmo de inicialização dos parâmetros.

CHAPTER 4

# Proposed Method

Although a simple static signature can be generated linking the points of the dynamic trajectory, additional dynamic information could be used to enrich the static signature. This addition is expected to lead more realistic and more discriminative images. Figure 14 shows how the proposed approach is used to generate a static signature based on online data.

In this chapter, we describe our proposed approach. First, the dataset used to train our model is described. Afterwards, we describe how the data was preprocessed. Finally, the CNN model is detailed.

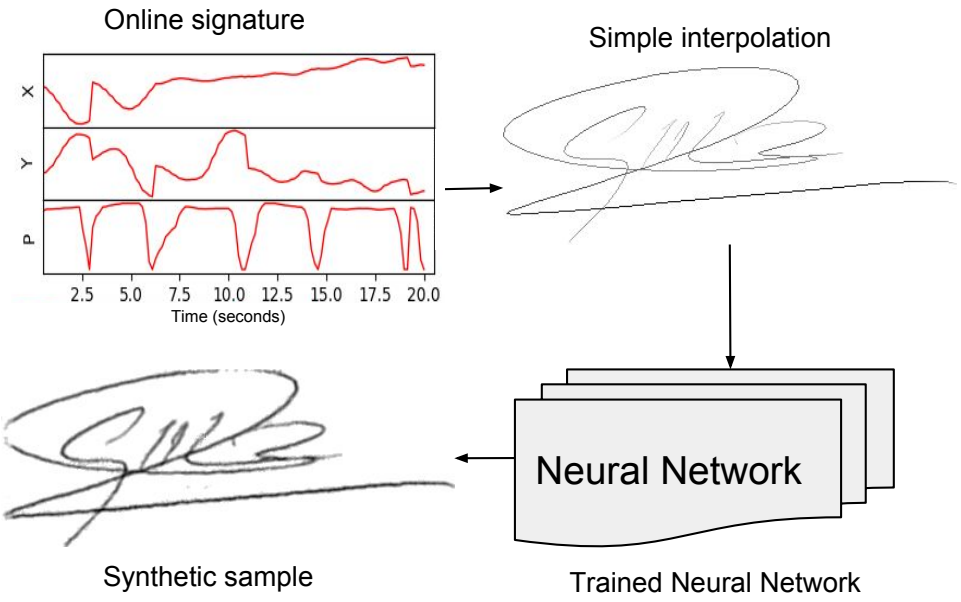



Figure 14 – The proposed approach diagram and an example of the synthetic signature generation.

## 4.1 Online to offline training data

In order to train our Neural Network to perform the approximation task of “online to offline conversion” we need both the online version of a handwritten manuscript containing the trajectory and pressure information mapped to the respective resulting offline representation, as in Figure 3. In order to acquire the online trajectory as well as the digital image for the same handwriting signal, some points have to be considered.

online signatures can’t be projected in the respective offline version

don’t match perfectly if you plot both in a single image (like your "onoff.png"). I’m not sure how they collected BiosecurID but this is a difficult problem to solve since BiosecurID was collected a long time ago in ATVS

To the best of our knowledge the creators of the BiosecurID had not this characteristic in mind as well as the creators of other [46, 47, 19, 48, 49, 50] publicly available dual modal signature datasets.

Figure 15 – The proposed approach diagram and an example of the synthetic signature generation.

For each sample in the database, two complementary files are available. One of them, produced during the online acquisition, contains the list of the coordinates of the pen trajectory, the other one is the digital image of this piece of handwriting produced by the scanner. Of course, these two types of information should be available within the same coordinate system, with the same origin, the same resolution and orientation. But, as the two acquisition systems (tablet and scanner) are processing the writing separately and with their own parameters, this assumption is not satisfied directly. Geometrical transformations have to be applied to compensate for these differences.

The neural network was trained using the data from the IRONOFF dataset [4]. The database contains a total of 23000 mapped online and offline information of the handwriting manuscripts. The offline handwriting signals have been sampled with a spatial resolution of 300 dots per inch (DPI), with 8 bits per pixel (256 gray level). The training data set contains 24,177 word images, and another 12,219 images have been used as test data.

Figure 16 – An offline manuscript mapped with the respective online trajectory. Image extracted from [4].

## 4.2 Preprocessing

The neural networks expect inputs of a fixed size, where signatures vary significantly in shape (in IRONOFF, they range from small handwritten samples of size 167x214 to larger samples of size 548x215 pixels). In order to have a fixed size, we first normalize the images to the largest image size, by padding the images with white background. In this case, we centered the manuscript in a canvas of size 548x215 pixels, aligning the center of mass of the sample to the center of the image, similar to previous approaches in the literature, e.g. [51]. We then rescaled the images to 383x150 pixels, i.e. 70% of the canvas size, maintaining the aspect ratio of the original sample. This size was chosen to be large enough to keep details from the pen strokes in the manuscript.

Besides resizing the images to a standard size, we also performed the following pre-processing steps:

- Inverted the images: we inverted the images so that the white background corresponded to pixel intensity 0.
- Normalized the input: we normalized the input to the neural network by dividing each pixel by the standard deviation of all pixel intensities. We do not normalize the data to have mean 0 (another common pre-processing step) since we want the background pixels to be zero-valued.
- Interpolation: The online sequences $(x_t, y_t, p_t)$ are linearly interpolated using the Bresenham line algorithm [52] to obtain 8-connected sequences.

Figure 17 shows an example of the preprocessed input and the groundtruth that are being fed to the neural network during training phase.

## 4.3 Neural Network model

The fully convolutional neural network (FCN) is adopted to learn an end- to-end nonlinear mapping from online represetation to static image.

Our model is based on the concept of autoassociative memory [53, 54], commonly known as autoencoders. Autoencoders are normally used to reduce high-dimensional data into a

(a) input (b) ground truth

Figure 17 – Preprocessed data used during the training phase. (a) is the interpolated online sample, used as input and (b) is the expected prediction from the neural network, the ground truth.

low-dimensional code to later reconstruct the data from the compressed code, i.e., the desired output is the same of the input pattern [55, 56]. In our case, instead of expect the output to be the same as the input, our Neural Network model is fed with the interpolated online sample pressure information and we expect the respective offline sample on the output, i.e. the input is the pressure information and the desired output is the static manuscript.

[57] we can remove each pooling layer and increase the stride of the convolutional layer that preceded it accordingly

We use an architecture based on the Convolutional Autoencoder [58]. The expectation is that by learning to transform between online information to offline manuscripts, the network will learn convolution filters that are relevant to synthesize offline signatures based the online sample.

Table 1 – Summary of the CNN Layers

| **Layer** | **Size** |
|---|---|
| Convolution | 16x3x3 |
| Convolution | 32x3x3 |
| Convolution | 32x3x3 |
| Convolution | 64x3x3 |
| Transpose Convolution | 64x3x3 |
| Transpose Convolution | 32x3x3 |
| Transpose Convolution | 32x3x3 |
| Transpose Convolution | 16x3x3 |

We used the convolutions architecture similar to the one proposed on [59] and the transposed convolutions were symmetric to the convolution operations. Some tests showed that the capacity of this network seems to be too large for the problem at hand, particularly considering the amount of convolution operations. We observed that the Neural Network was not learning useful transformations on the first few epochs. We obtained better results with. For the purpose of replicating our experiment, we provide a full list of the parameters used in our tests. Table **??** lists the definition of the Convolutional Autoencoder layers. For convolution and

transpose convolution layers, we list the size as NxHxW where N is the number of filters, H is the height and W is the width of the convolution and transpose convolution windows, respectively.

We used Leaky Rectified Linear Units (LReLUs) the activation function for all convolutional layers. We also tried the Rectified Linear Units (ReLUs) but we observed that on the first 10 epochs the Neural Network seemed to be converging faster with the LReLUs.

For every convolutional layer we use a stride (the distance between applications of the convolution operation) of 2x2, and we pad the input for every layer with 0 evenly left and right. We initialize the weights of the model using the technique proposed by Glorot and Bengio [45], and the biases to 0. We trained the model with Adam optimizer to minimize the minimum squared error (MSE) loss for 100 epochs, using a learning rate of 0.001, and mini-batches of size 16. The network was trained using the library Tensorflow, and took around 72 hours to train on a GTX 670 GPU.

## 4.4 Training Results

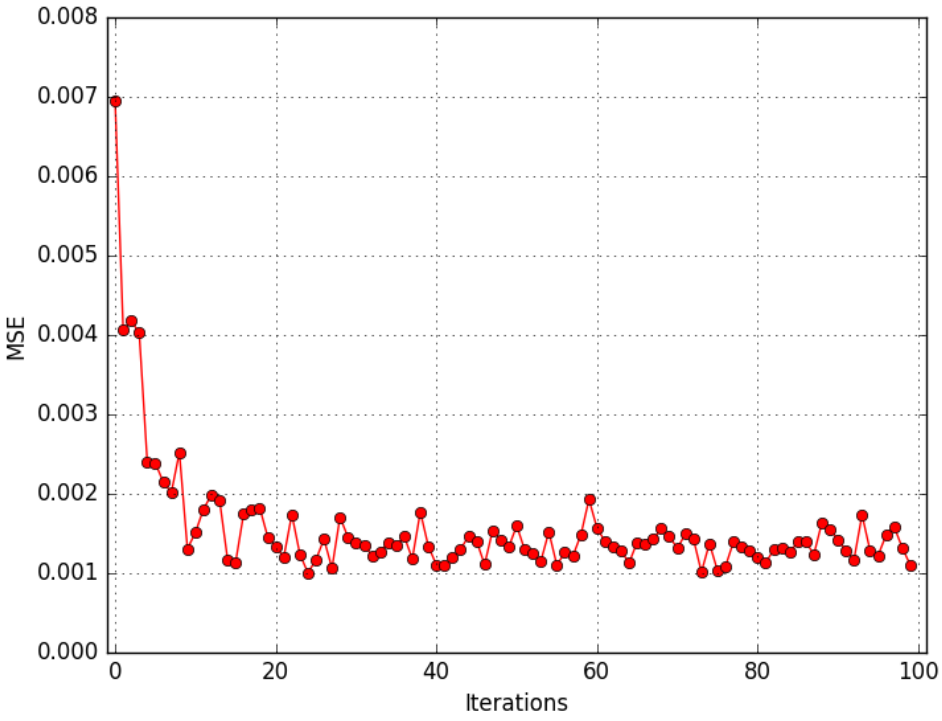
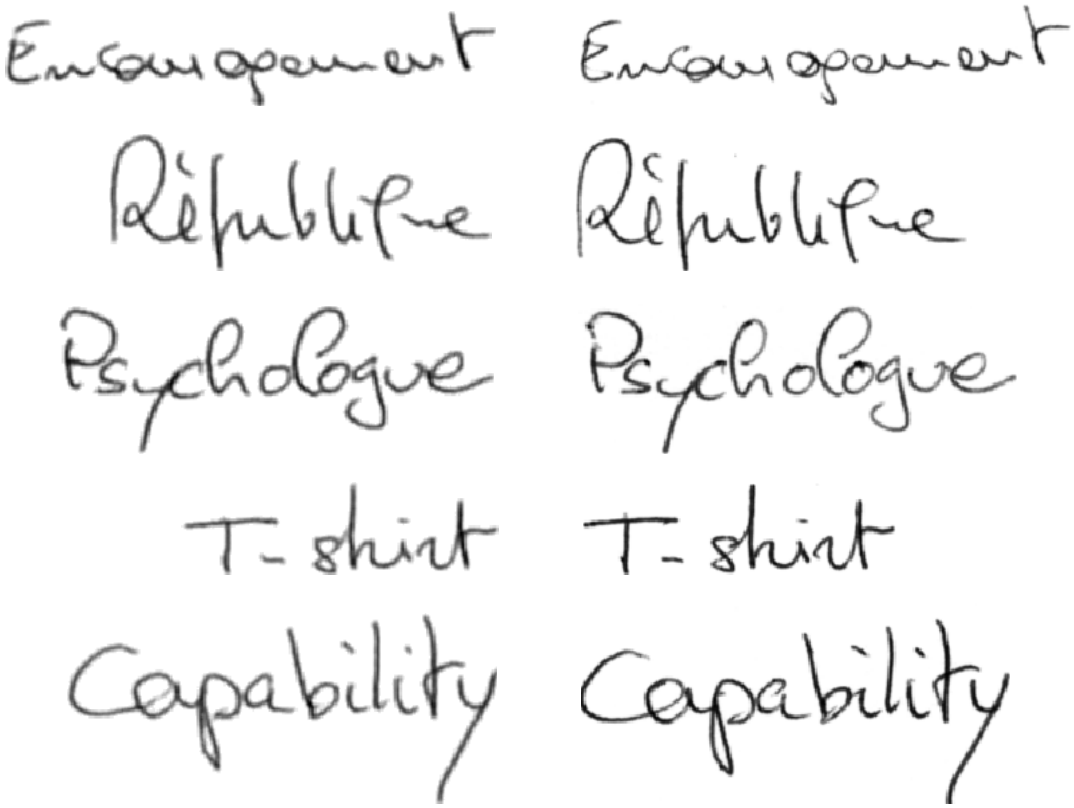



Figure 18 – Iterations versus MSE



(a) Synthetic Handwritten Manuscript

(b) Expected Real Manuscript - Groundtruth

Figure 19 – Not cherry-picked

CHAPTER 5

# Experimental Evaluation

In order to evaluate the quality of the synthetic signatures generated by our system we follow the same protocol presented on the work of Diaz [10]. Namely we use a state-of-the-art offline verification system and a dataset comprising both online and offline signatures in order to train the verification system to evaluate the synthetic signatures.

The goal of the experiments is to measure the quality of the synthetic signatures taking into account an offline verification system performance. The questions raised are (i) is the synthetic signatures system performance similar to the real offline signatures performance? (ii) is the performance even if the enrollment protocol changes? (iii) is it feasible to increase the number of samples at the enrollment stage with synthetic signatures?

## 5.1 Off-line signature verification system

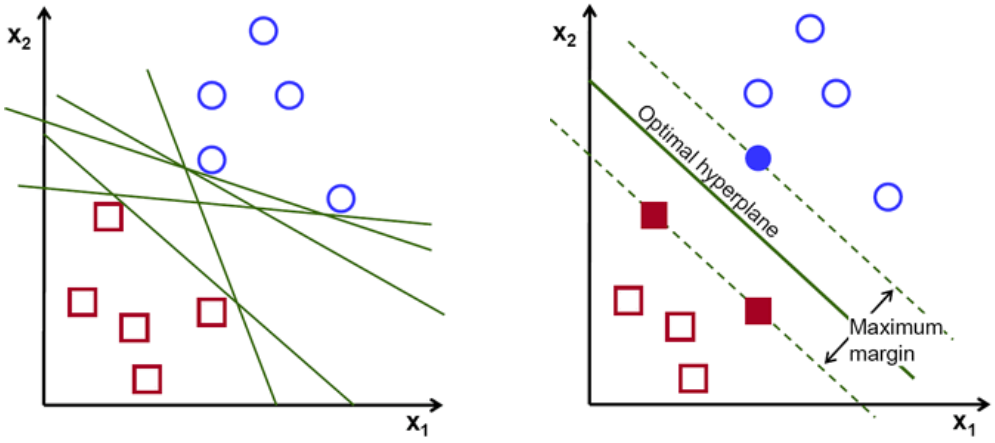
The system used for the evaluation of the real and synthetic signatures is a Linear SVM classifier and is based on a state of the art feature extraction approach [60]. The feature extraction system [4] uses ideas from transfer learning and multi-task learning to learn features using Convolutional Neural Networks (CNN).

### 5.1.1 Support Vector Machine

SVMs represent a special class of linear classifiers. In order to classify a pattern as belonging to one of two classes, an SVM constructs a hyperplane in the feature space, such

[4] https://www.etsmtl.ca/Unites-de-recherche/LIVIA/Recherche-et-innovation/Projets/Signature-Verification

that it maximally separates the margin between the two classes. For this reason, SVMs are also referred to as maximum margin classifiers.




(a) mono-session - random forgeries (b) mono-session - skilled forgeries

Figure 20 – Boxplot comparison for running 30 times the Experiment 1 - mono-session scenario and multi-session. (a) mono-session scenario, random forgeries, (b) mono-session scenario, skilled forgeries, (c) multi-session scenario, random forgeries, (d) multi-session scenario, skilled forgeries.

### 5.1.2 Feature Learning

ideas from transfer learning and multi-task learning to learn features using Convolutional Neural Networks (CNN).

## 5.2 Evaluation Database

The evaluation experiments were carried out on the BiosecurID database [61]. This multimodal database was made publicly available containing signatures of 132 subjects. Signatures were captured using a special digital inking pen on a paper placed over a digitizing tablet. Consequently, both versions, online and offline, of the exact same real signature were acquired at the same time. This characteristic, therefore, makes BiosecurID the ideal benchmark for the experimental evaluation conducted in this work.

The signatures were captured in 4 sessions distributed over 4 months. Each subject signed 4 times and forged 3 signatures per session, thus leading to each subject having 4 genuine signatures x 4 sessions = 16 genuine samples and 3 signature forgeries x 4 sessions = 12 skilled forgeries.

MCYT This dataset includes 75 signers collected at four different Spanish universities. The corpus includes 15 genuine and 15 deliberately forged signatures acquired in two sessions. All the signatures were acquired with the same inking pen and the same paper templates. The paper templates were scanned at 600 dpi with 256 grey levels.

## 5.3 Experimental Protocol

In order to answer the questions stated at the beginning of this section and accomplish a fair comparison of our work and the state of the art, we follow the same experiment protocol proposed by Diaz [10]. Two different experiments are carried out. The Experiment 1 focus on evaluating the synthetic signatures performance in comparison to real signatures and the Experiment 2 evaluates the feasibility of synthetically increasing the number of samples available in a dataset.

For both experiments the BiosecurID dataset is split into two subsets. The first 90 users are separated as the enrollment set, used to compute the genuine and skilled impostor scores. The remaining 42 authors are considered as the test set and are used to compute the random impostor scores. The performance is evaluated in terms of equal error rate (EER), which is the point in the Detection Error Tradeoff curve (DET) where the false acceptance rate equals the false rejection rate.

### 5.3.1 Experiment 1

Two different protocols have been considered to compute the 90 user enrolled models: (i) a mono-session approach, uses the four samples of the first acquisition session (ii) a multi-session approach, uses one sample of each of the 4 acquisition sessions.

In both cases, genuine scores are computed matching the non-enrolled genuine samples of the subject (12) to the enrolled model (90×12 = 1080 genuine scores). Random impostor scores are calculated comparing the first sample of the test subjects to the enrolled model, leading to 90×42 = 3780 random impostor scores, and skilled impostor scores are calculated with the skilled forgeries samples of the enrolled users (12 per subject) to the enrolled model (90×12 = 1080 skilled impostor scores).

### 5.3.2 Experiment 2

This experiment is designed to assess whether synthetically increasing the enrollment dataset leads to a better recognition performance. Three different enrollment sets are considered in this experiment:

- 4 real samples belonging to the first acquisition session
- 8 real samples belonging to the first and the second sessions
- 4 real samples belonging to the first session plus 4 synthetic samples belonging to the second session.

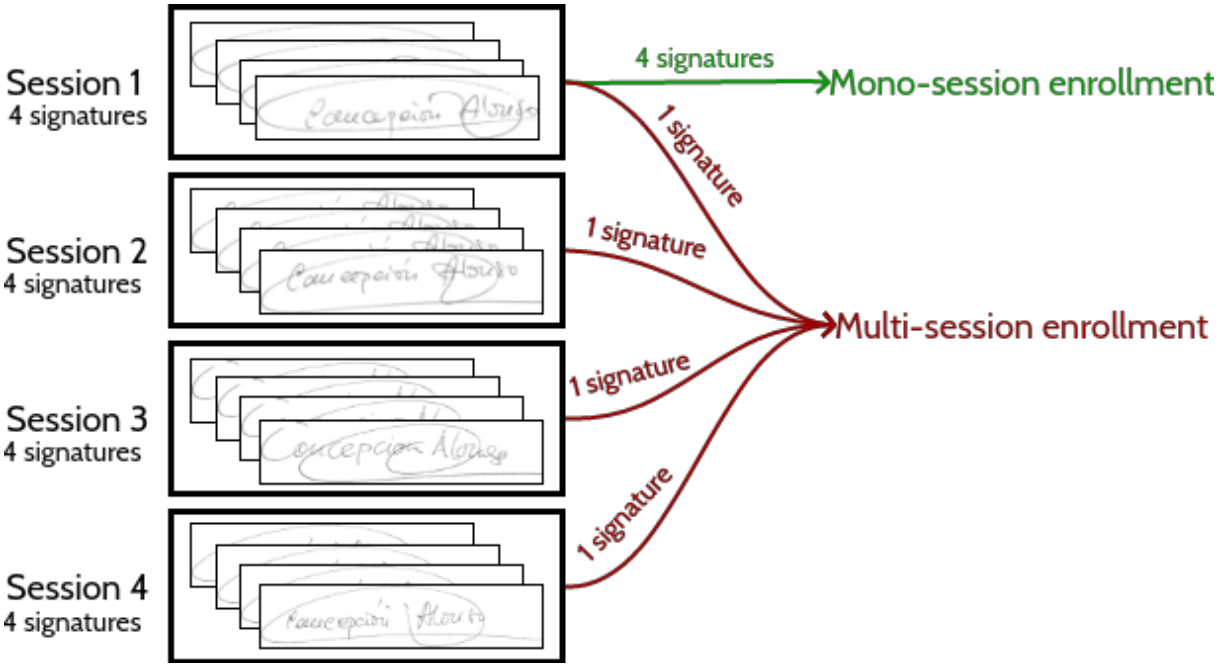



Figure 21 – The proposed approach diagram and an example of the synthetic signature generation.

## 5.4 Performance Assessment

AHSVS efficiency is quantitatively measured by two rates: False Rejection Rate (FRR) which is the percentage of genuine signatures treated as forgeries, and False Acceptance Rate (FAR) which is the percentage of forged signatures treated as genuine.

A derived metric usually reported is the Average Error rate (AER) which is the average of FAR and FRR. Moreover, when experimenting an AHSVS, the trade-off between FRR and FAR must be taken into account based on the type of application and other aspects related to where the system is used. When the decision threshold of a system is set to have the FRR approximately equal to the FAR, the Equal Error Rate (EER) is calculated.

Besides quantitative results, the performance can be also compared and visualized using graphs. Receiver operating characteristics (ROC) graphs have been used increasingly in machine learning and data mining research [62]. Inicialmente, a curva ROC foi desenvolvida para demonstrar as relações sinal-ruído, interpretando o sinal como verdadeiros positivos (sensibilidade) e o ruído, os falsos positivos (especificidade). Assim, a leitura da curva ROC é de um gráfico de sensibilidade ou taxas de verdadeiros positivos versus taxa de falsos positivos, como apresentado na Figura 26. Os gráficos ROC são bidimensionais, estando o eixo Y com os valores verdadeiros positivos e o eixo X preenchido com os valores da taxa falsos positivo.

O objetivo da ferramenta curva ROC é atingir a representação perfeita do experimento, considerando o conjunto de amostras analisado. O objeto de estudo tem sua avaliaçãoidentificada pela localidade do ponto no gráfico. Quão mais próximo o ponto estiver do eixo Y, melhor será o resultado. Assim, a diagonal traçada do ponto (0,0) ao ponto (1,1) serve de direcionamento para

identificar quais são os melhores resultados, estes localizados predominantemente acima da reta [FAWCELT, 2006].

A métrica utilizada no estudo do ICDAR 2009 [BLANKERS et al, 2009] foi a Detection Error Trade-off curve (DET) [MARTIN, 1997] (gráfico visualizado na Figura 28). Esta métrica é mais conhecida como a curva DET. Esta curva corresponde a um gráfico que estabelece uma relação entre as taxas de erro diferentemente da curva ROC. A curva é definida como a linha contendo os pontos onde x = y, ou seja, posição onde os erros FRR e FAR possuem o mesmo valor. Assim, o gráfico é gerado baseando-se na taxa EER. No gráfico DET o melhor resultado está localizado o mais próximo do eixo inicial (o ponto 0 (zero)). O mais próximo apresenta, assim, o menor erro de classificação. Esta métrica foi utilizada pelo ICDAR para definir o algoritmo de classificação da competição com o melhor EER nos problemas de verificação de assinaturas nos modos off-line e on-line.

## 5.5 Statistical Evaluation

CHAPTER 6

# Results

As described in Chapter 5, two different experiments are carried out in order to analyze the quality of the synthetic signatures and the feasibility of synthetically increasing the enrollment set samples. The goal of the experiments is threefold, namely: (i) measure the similarity between real and synthetic images (ii) assess whether using synthetic signatures affects an offline signature verification system recognition performance (iii) analyze the feasibility of using real and synthetic signatures on the enrollment set.

We compare our results with the state-of-the-art, particularly the approach proposed by Diaz [10]. Specifically, our proposed method is compared with the "Image enhanced" synthetic signatures made available as part of the BiosecurID [61] dataset. The reported EER is achieved for both approaches on the same experimental conditions.

The recognition rates of three types of offline signatures are reported: (i) real offline signatures, which are the corresponding real offline samples of the online signatures used to generate the synthetic samples, they serve us as a ground-truth, i.e. the ideal synthetic signature should be similar to it (ii) synthetic signatures generated with the method proposed by Diaz [10] (iii) our proposed method synthetic signatures.

## 6.1 Synthetic signatures in comparison to real signatures - Experiment 1

In order to evaluate the performance of the synthetic signature signatures described in Chapter 4 and the state-of-the-art, two different protocols are followed:

- mono-session: signatures from the first session on the enrollment set
- multi-session: one signature per session.

Table 2 – EER for real, synthetic samples from Diaz [10] and our proposed method synthetic offline signatures, for all the approaches considered under the two possible scenarios (i.e., random and skilled forgeries)

| **Mode** | **Skilled Forgeries** | | |
|---|---|---|---|
| | **Real** | **Diaz** | **Proposed method** |
| **mono-session** | 20.28% | 23.19% | 18.38% |
| **multi-session** | 17.59% | 22.27% | 16.48% |
| | **Random Forgeries** | | |
| | **Real** | **Diaz** | **Proposed method** |
| **mono-session** | 9.07% | 10.65% | 9.99% |
| **multi-session** | 5.60% | 10.00% | 6.48% |

Table 2 shows the EER achieved by the real and the synthetic signatures databases. As it may be observed, under the random forgeries scenario, the EERs achieved by the real and both types of synthetic signatures are very close. On the other hand we observe that under the skilled forgeries our proposed method synthetic signatures EERs yields better performance than both the real dataset and the synthetic signatures generated by Diaz.

In Figure 22 (left - mono-session, right - multi-session) we may observe that the behaviour of the system with real (gray dotted) and our proposed method synthetic signatures is quite similar, regardless of the training protocol or the scenario considered. Nevertheless, our proposed method synthetic samples show a bigger discriminative power for skilled forgeries.

Table 3 – EER for real, synthetic samples from Diaz [10] and our proposed method synthetic offline signatures for the Experiment 2 under the two possible scenarios, i.e., random (RF) and skilled forgeries (SF)

| **Genuine Training** | **SF** | **RF** |
|---|---|---|
| **4 real samples** | 21.55% | 10.26% |
| **4 real + 4 real samples** | 19.72% | 7.63% |
| **4 real + 4 synthetic from Diaz** | 24.19% | 7.72% |
| **4 real + 4 synthetic from the Proposed Method** | 19.17% | 9.74% |

Moreover, in Figure 23 we can see that when running the experiments 30 times, the results are consistent, with a low dispersion around the average. We can notice that our proposed method results are comparable to the real signatures.

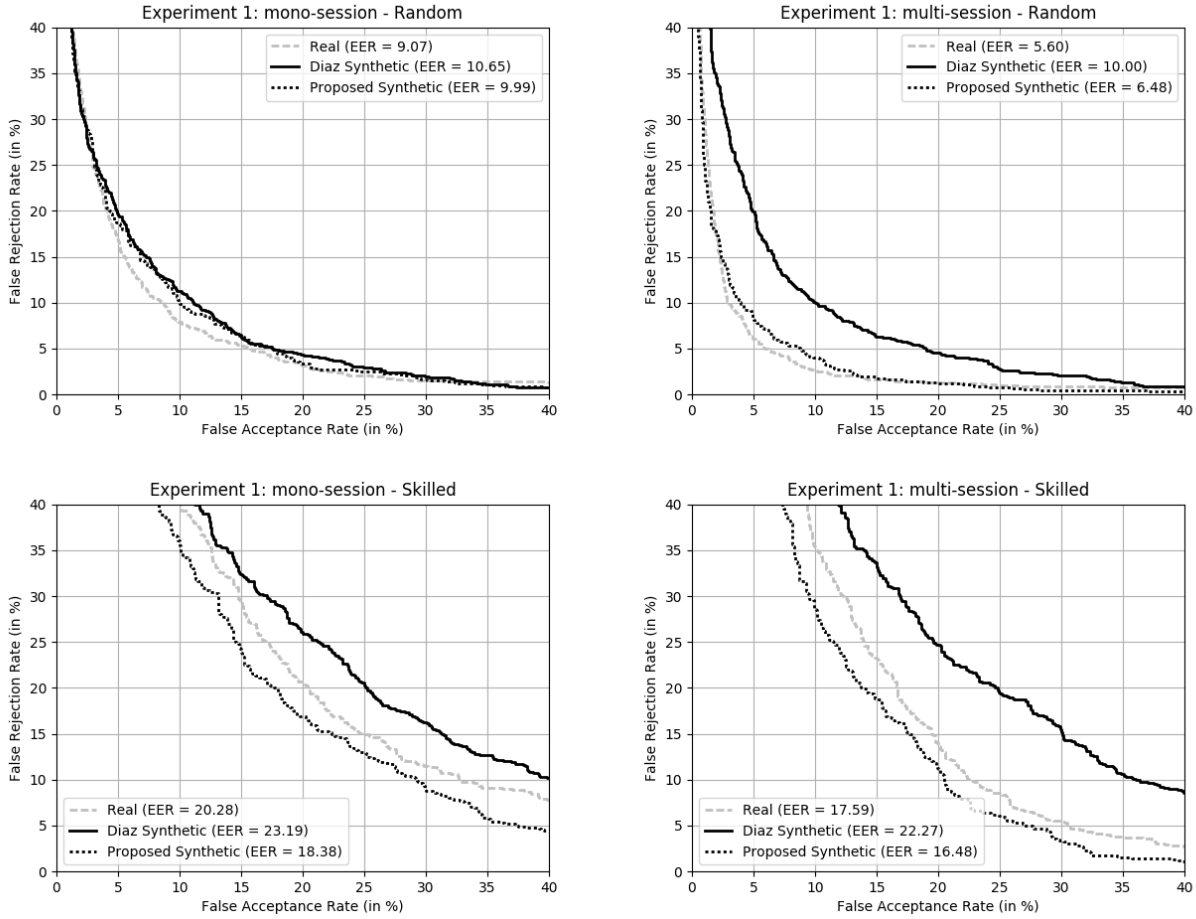
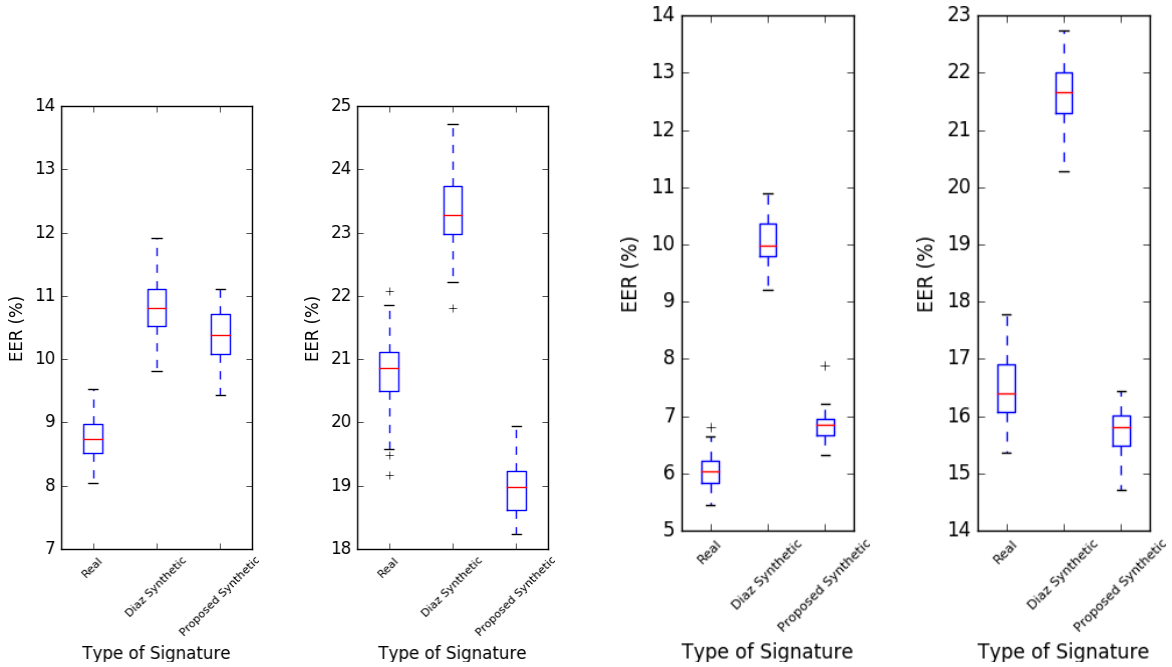



Figure 22 – DET curves for real offline signatures and synthetic signatures (from Diaz and our proposed method), for the first experiment (mono-session and multi-session enrollment), for the two scenarios considered (random and skilled impostors)

## 6.2 Feasibility of synthetically increasing the enrollment samples - Experiment 2

In the last experiment, the feasibility of synthetically increasing the signature verification enrollment set is analyzed. As it is described in Chapter 5, three different enrollment sets are considered. As it may be observed in Figure 24, the DET curves for the mixed enrollment (real + synthetic, grey dashed line), show a better performance compared to the case with only four real enrolled samples (black line), regardless of the operating point or the scenario considered. More specifically, the EER decreases from 10.26% to 9.74% on the random forgeries scenario and from 21.55% to 19.17% for skilled forgeries, respectively. The addition of synthetic samples for training thus leads to better recognition results.

It should also be noted that the behavior of the mixed enrollment is similar to the scenario with eight real enrolled samples (i.e., using eight real samples instead of four real and four synthetic, black line) for skilled forgeries, and even yields a small improvement on the skilled forgery recognition rates.




(a) mono-session - random forgeries (b) mono-session - skilled forgeries (c) multi-session - random forgeries (d) multi-session - skilled forgeries

Figure 23 – Boxplot comparison for running 30 times the Experiment 1 - mono-session scenario and multi-session. (a) mono-session scenario, random forgeries, (b) mono-session scenario, skilled forgeries, (c) multi-session scenario, random forgeries, (d) multi-session scenario, skilled forgeries.

Furthermore, we can see in Figure 25 that when running the experiments for 30 times, the results of our proposed method has a consistent mean, with a lower variance around the average. In addition to achieving a comparable EER to using 8 real samples on the enrollment set on the random forgery and skilled forgery scenario.

We may thus conclude that, our proposed system can be a good alternative to synthetically increase the offline signatures samples when complementary online samples are available in order to increase the accuracy of the offline verification system.

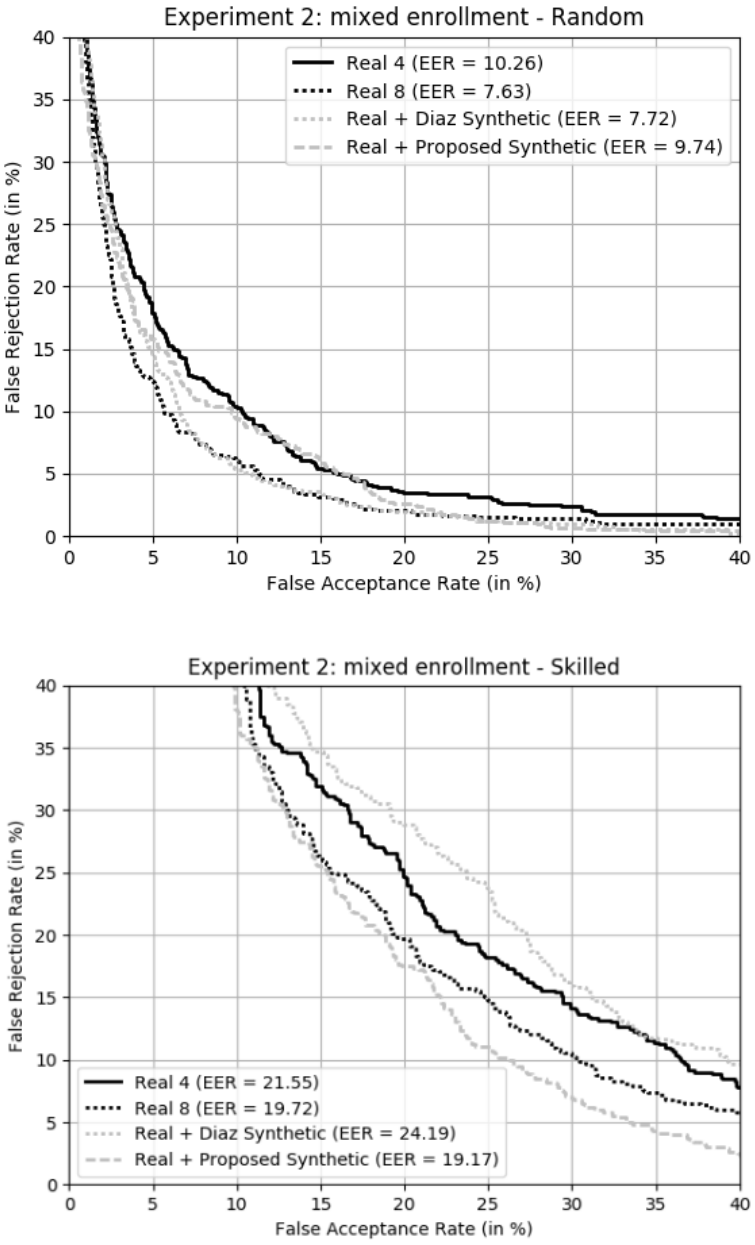
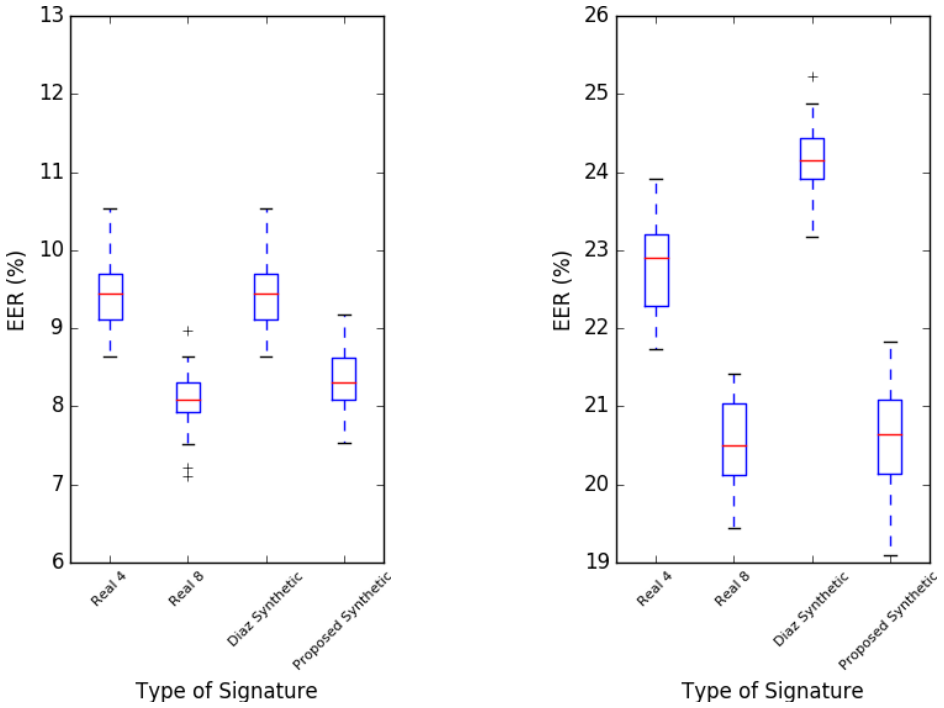



Figure 24 – DET curves for real offline signatures and synthetic signatures (from Diaz and our proposed method), for the second experiment, for the two scenarios considered (random and skilled impostors)




(a) mixed enrollment - random forgeries

(b) mixed enrollment - skilled forgeries

Figure 25 – Boxplot comparison for Experiment 2: mixed enrollment - (a) random forgeries and (b) skilled forgeries.

CHAPTER 7

# Conclusion and Future Works

A novel method to synthesize off-line signatures from dynamic information has been proposed. We describe our Neural Network approach that learns how to transform online manuscripts into static samples taking advantage of the data present in the IRONOFF dataset during the training phase of Neural Network.

We observe that our proposed synthetic images offer better performance when compared to the state-of-the-art and comparable performance to real signatures. We show that the synthetic signatures present high discriminative power when used to increase the enrollment set under the skilled forgeries scenario, one of the biggest challenges in signature verification.

We intend to explore in future works the combination of real online and synthetically generated offline signatures from our method, when only the online information is available, towards improved recognition results.

In this paper, we show that it is possible to model the "online to offline signature" conversion as a learning task.

Our Deep Neural Network model learns how the online biometric data must be used to build the grayscale signature image.

The proposed system can be combined with any other offline signature system and is a writer independent approach.

The proposed method synthesizes offline signatures comparable to real signatures.

Our synthetic signatures can be used to increase the enrollment set, leading to improved EER, inclusive to skilled forgeries.

# Bibliography

[1] JAIN, A. K.; ROSS, A.; PRABHAKAR, S. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, IEEE, v. 14, n. 1, p. 4–20, 2004.

[2] IMPEDOVO, D.; PIRLO, G. Automatic signature verification: the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 38, n. 5, p. 609–635, 2008.

[3] FIERREZ, J.; ORTEGA-GARCIA, J. On-line signature verification. In: *Handbook of biometrics*. [S.l.]: Springer, 2008. p. 189–209.

[4] VIARD-GAUDIN, C.; LALLICAN, P. M.; KNERR, S.; BINTER, P. The ireste on/off (ironoff) dual handwriting database. In: IEEE. *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*. [S.l.], 1999. p. 455–458.

[5] HERRERO, J. G.; FIERREZ, J.; DÍAZ, M. M.; ORTEGA-GARCIA, J. Synthetic generation of handwritten signatures based on spectral analysis. In: SOCIETY OF PHOTO-OPTICAL INSTRUMENTATION ENGINEERS. *Proceedings of SPIE-The International Society for Optical Engineering*. [S.l.], 2009.

[6] GALBALLY, J.; PLAMONDON, R.; FIERREZ, J.; ORTEGA-GARCIA, J. Synthetic on-line signature generation. part i: Methodology and algorithms. *Pattern Recognition*, Elsevier, v. 45, n. 7, p. 2610–2621, 2012.

[7] FERRER, M. A.; DIAZ-CABRERA, M.; MORALES, A. Synthetic off-line signature image generation. In: IEEE. *Biometrics (ICB), 2013 International Conference on*. [S.l.], 2013. p. 1–7.

[8] FERRER, M. A.; DIAZ-CABRERA, M.; MORALES, A.; GALBALLY, J.; GOMEZ-BARRERO, M. Realistic synthetic off-line signature generation based on synthetic on-line data. In: IEEE. *Security Technology (ICCST), 2013 47th International Carnahan Conference on*. [S.l.], 2013. p. 1–6.

[9] UPPALAPATI, D. Integration of Offline and Online Signature Verification systems. *Department of Computer Science and Engineering, IIT, Kanpur*, 2007.

[10] DIAZ-CABRERA, M.; GOMEZ-BARRERO, M.; MORALES, A.; FERRER, M. A.; GALBALLY, J. Generation of enhanced synthetic off-line signatures based on real on-line data. In: IEEE. *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. [S.l.], 2014. p. 482–487.

[11] MELO, V. K. S. L.; BEZERRA, B. L. D.; NASCIMENTO, R. H. S. N. do; MOURA, G. C. D. de; MARTINS, G. L. L. d. S.; PIRLO, G.; IMPEDOVO, D. Datasets for Handwritten Signature Verification: A Survey and a New Dataset, the RPPDI-SigData. In: *Handwriting:*

*Recognition, Development and Analysis, 2017*. [S.l.]: Nova Science Publishers, 2017. cap. 14, p. 345–361.

[12] JAIN, A.; HONG, L.; PANKANTI, S. Biometric identification. *Communications of the ACM*, ACM, v. 43, n. 2, p. 90–98, 2000.

[13] ROSS, A.; NANDAKUMAR, K.; JAIN, A. K. Introduction to multibiometrics. In: *Handbook of Biometrics*. [S.l.]: Springer, 2008. p. 271–292.

[14] JAIN, A. K.; NANDAKUMAR, K.; ROSS, A. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, Elsevier, v. 79, p. 80–105, 2016.

[15] PAL, S.; PAL, U.; BLUMENSTEIN, M. *Signature-Based Biometric Authentication.* 2014.

[16] GUPTA, G.; MCCABE, A. A review of dynamic handwritten signature verification. *James Cook Univ., Australia*, 1997.

[17] WACOM. *STU-500*. 2016. Disponível em: <http://www.wacom.com/en-us/enterprise/business-solutions/hardware/signature-pads/stu-500>.

[18] NEL, E.-M.; PREEZ, J. A. D.; HERBST, B. M. Estimating the pen trajectories of static signatures using Hidden Markov models. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 27, n. 11, p. 1733–1746, 2005.

[19] HEUVEL, C. van den; FRANKE, K.; VUURPIJL, L. The ICDAR 2009 signature verification competition. *ICDAR2009 proceedings*, 2009.

[20] PIRLO, G.; IMPEDOVO, D.; FAIRHURST, M. *Advances in Digital Handwritten Signature Processing: A Human Artefact for E-society*. [S.l.]: World Scientific, 2014.

[21] PLAMONDON, R.; LORETTE, G. Automatic signature verification and writer identification—the state of the art. *Pattern recognition*, Elsevier, v. 22, n. 2, p. 107–131, 1989.

[22] HAFEMANN, L. G.; SABOURIN, R.; OLIVEIRA, L. S. Offline handwritten signature verification-literature review. *arXiv preprint arXiv:1507.07909*, 2015.

[23] ORTEGA-GARCIA, J.; FIERREZ-AGUILAR, J.; SIMON, D.; GONZALEZ, J.; FAUNDEZ-ZANUY, M.; ESPINOSA, V.; SATUE, A.; HERNAEZ, I.; IGARZA, J.-J.; VIVARACHO, C. et al. MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings-Vision, Image and Signal Processing*, IET, v. 150, n. 6, p. 395–401, 2003.

[24] BEZERRA, B. L. D.; ZANCHETTIN, C.; TOSELLI, A. H.; PIRLO, G. *Handwriting: Recognition, Development and Analysis* . [S.l.]: Nova Science Publishers, 2017.

[25] FAIRHURST, M. Signature verification revisited: promoting practical exploitation of biometric technology. *Electronics & communication engineering journal*, IET, v. 9, n. 6, p. 273–280, 1997.

[26] GUEST, R. M.; HURTADO, O. M.; HENNIGER, O. Assessment of methods for image recreation from signature time-series data. *IET biometrics*, IET, v. 3, n. 3, p. 159–166, 2013.

[27] RABASSE, C.; GUEST, R. M.; FAIRHURST, M. C. A new method for the synthesis of signature data with natural variability. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 38, n. 3, p. 691–699, 2008.

[28] DIAZ-CABRERA, M.; FERRER, M. A.; MORALES, A. Cognitive inspired model to generate duplicated static signature images. In: IEEE. *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. [S.l.], 2014. p. 61–66.

[29] HAYKIN, S. S.; HAYKIN, S. S.; HAYKIN, S. S.; HAYKIN, S. S. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, NJ, USA:, 2009. v. 3.

[30] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.

[31] MYNSKI, M.; PAPERT, S. Perceptrons: An introduction to computational geometry. *MA: MIT Press, Cambridge*, 1969.

[32] BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006.

[33] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks*, Elsevier, v. 2, n. 5, p. 359–366, 1989.

[34] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation*. [S.l.], 1985.

[35] BENGIO, Y. et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, Now Publishers, Inc., v. 2, n. 1, p. 1–127, 2009.

[36] LECUN, Y.; BENGIO, Y. et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, v. 3361, n. 10, p. 1995, 1995.

[37] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Research, v. 521, n. 7553, p. 436–444, 2015.

[38] HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006.

[39] NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814.

[40] SRIVASTAVA, N.; HINTON, G. E.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, v. 15, n. 1, p. 1929–1958, 2014.

[41] LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

[42] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105.

[43] DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[44] LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 3431–3440.

[45] GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. [S.l.: s.n.], 2010. p. 249–256.

[46] GARCIA-SALICETTI, S.; BEUMIER, C.; CHOLLET, G.; DORIZZI, B.; JARDINS, J. L. L.; LUNTER, J.; NI, Y.; PETROVSKA-DELACRÉTAZ, D. BIOMET: a multimodal person authentication database including face, voice, fingerprint, hand and signature modalities. In: SPRINGER. *International Conference on Audio-and Video-based Biometric Person Authentication*. [S.l.], 2003. p. 845–853.

[47] DUMAS, B.; PUGIN, C.; HENNEBERT, J.; PETROVSKA-DELACRÉTAZ, D.; HUMM, A.; EVÉQUOZ, F.; INGOLD, R.; ROTZ, D. V. MyIDea-multimodal biometrics database, description of acquisition protocols. *Proc. Third COST*, v. 275, p. 59–62, 2005.

[48] AHMAD, S. M. S.; SHAKIL, A.; AHMAD, A. R.; BALBED, M. A. M.; ANWAR, R. M. SIGMA-A Malaysian signatures' database. In: IEEE. *2008 IEEE/ACS International Conference on Computer Systems and Applications*. [S.l.], 2008. p. 919–920.

[49] MALIK, M. I.; LIWICKI, M.; ALEWIJNSE, L.; OHYAMA, W.; BLUMENSTEIN, M.; FOUND, B. ICDAR 2013 Competitions on Signature Verification and Writer Identification for On- and Offline Skilled Forgeries (SigWiComp 2013). In: IEEE. *2013 12th International Conference on Document Analysis and Recognition*. [S.l.], 2013. p. 1477–1483.

[50] MALIK, M. I.; AHMED, S.; MARCELLI, A.; PAL, U.; BLUMENSTEIN, M.; ALEWIJNS, L.; LIWICKI, M. ICDAR2015 competition on signature verification and writer identification for on-and off-line skilled forgeries (SigWIcomp2015). In: IEEE. *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. [S.l.], 2015. p. 1186–1190.

[51] POURSHAHABI, M. R.; SIGARI, M. H.; POURREZA, H. R. Offline handwritten signature identification and verification using contourlet transform. In: IEEE. *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of*. [S.l.], 2009. p. 670–673.

[52] BRESENHAM, J. E. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, v. 4, n. 1, p. 25–30, 1965. ISSN 0018-8670.

[53] JENSEN, O.; IDIART, M.; LISMAN, J. E. Physiologically realistic formation of autoassociative memory in networks with theta/gamma oscillations: role of fast nmda channels. *Learning & Memory*, Cold Spring Harbor Lab, v. 3, n. 2-3, p. 243–256, 1996.

[54] WEINBERGER, N. M. Specific long-term memory traces in primary auditory cortex. *Nature reviews. Neuroscience*, Nature Publishing Group, v. 5, n. 4, p. 279–290, 2004.

[55] KRAMER, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, Wiley Online Library, v. 37, n. 2, p. 233–243, 1991.

[56] HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *science*, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.

[57] SPRINGENBERG, J. T.; DOSOVITSKIY, A.; BROX, T.; RIEDMILLER, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[58] MASCI, J.; MEIER, U.; CIREŞAN, D.; SCHMIDHUBER, J. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning–ICANN 2011*, Springer, p. 52–59, 2011.

[59] SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[60] HAFEMANN, L. G.; SABOURIN, R.; OLIVEIRA, L. S. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, Elsevier, v. 70, p. 163–176, 2017.

[61] FIERREZ, J.; GALBALLY, J.; ORTEGA-GARCIA, J.; FREIRE, M. R.; ALONSO-FERNANDEZ, F.; RAMOS, D.; TOLEDANO, D. T.; GONZALEZ-RODRIGUEZ, J.; SIGUENZA, J. A.; GARRIDO-SALAS, J. et al. BiosecurID: a multimodal biometric database. *Pattern Analysis and Applications*, Springer, v. 13, n. 2, p. 235–246, 2010.

[62] FAWCETT, T. An introduction to ROC analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.

APPENDIX A

# Publications

As results of this work, a chapter has been published in the book Handwriting: Recognition, Development and Analysis and a paper has been submitted to the peer-reviewed journal Pattern Recognition Letters.

**Title:** Datasets for Handwritten Signature Verification: A Survey and a New Dataset, the RPPDI SigData.

**Authors:** Melo, V.K.S.L., Bezerra, B.L.D., do Nascimento, R.H.S.N., de Moura, G.C.D., Martins, G.L.L.d.S., Pirlo, G., Impedovo, D.

**In:** Handwriting: Recognition, Development and Analysis, ISBN 978-1-53611-957-2, Nova Science Publishers, New York, 2017

**Abstract:** As a result of a wide research in automatic handwritten signature verification, multiple data sets and competitions emerged to cover different tasks for online/offline signature verification and signature segmentation from scanned documents. The purpose of this work is to analyze and discuss the most used data sets in the literature in order to find what are the challenges pursued by the community in the past few years. The data sets, which were collected from recent publications and competitions, are described regarding its main characteristics such as the number of authors, signatures per author and the method used for the acquisition phase. We analyzed several databases used for different topics related to either online or offline signature verification. Since we did not find in this review a data set that can be used to explore signature verification considering different acquisition sources and writing conditions, specifically a data set consisting of dynamic signatures taken from multiple devices and static signatures with diverse signing area constraints and background textures, we introduce a new data set acquired from pen tablets, smartphones and scanned from different types of documents. Therefore, researchers can use this

new data set not only to investigate the multi-domain signature verification problem but to take into account in their solutions the signature segmentation problem in real world documents with complex backgrounds.

**Title:** A Deep Learning Approach to Generate Off-line Handwritten Signatures Based on On-line Samples

**Authors:** Melo, V.K.S.L., Bezerra, B.L.D., Impedovo, D., Pirlo, G.

**(In revision):** Pattern Recognition Letters

**Abstract:** One of the main challenges of offline signature verification is the absence of large databases. A possible alternative to overcome this problem is the generation of synthetic signature databases. In this work, a novel method for the generation of synthetic off-line signatures based on dynamic information is presented. In contrast to the state-of-the-art, we propose a synthesis approach under the perspective of supervised training, in which our learning model is trained to perform the task of "online signature to offline signature conversion". The proposed approach is based on a Deep Convolutional Neural Network trained to learn how online handwritten manuscripts of the IRONOFF dataset are transformed into the offline domain. The main goal of the proposed method is to synthetically enlarge existing offline signature datasets based on online signature samples towards an improvement on the recognition rates of offline signature verification systems. For these purposes, a machine-oriented evaluation on the BiosecurID signature dataset is carried out. Through the synthetic samples generated with the proposed method, we show a verification performance similar to the one offered by real authentic signatures and assuring improvements of the Equal Error Rate results in comparison with the current state-of-the-art method.