**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

Master's Degree Dissertation

Recife, august, 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

Master's Degree Dissertation

Dissertation presented to the Programa de Pós-Graduação acadêmico em ENGENHARIA DE COMPUTAÇÃO da Universidade de Pernambuco as a partial requisite to obtain the degree of Master of Science in Computer Engineering.

Prof. Dr. Byron Leite Dantas Bezerra
Advisor
Prof. Dr. Giuseppe Pirlo
Co-Advisor

Recife, august, 2017

à minha família.

Abstract

The task of offline signature verification typically presents the major problem of reduced training sets. In order to increase the number of samples, the use of synthetic signatures can be regarded as a possible alternative to overcome this problem. 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 off-line signature conversion". The proposed approach is based on a Deep Convolutional Neural Network trained to learn how online 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 when complementary online signatures are avaiable 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 our proposed method model can learn how the online biometric data must be used to build the grayscale signature image. Specifically, we observe that the synthetic samples generated with our proposed method are comparable to real signatures, achieving a verification performance similar to the one offered by real signatures. We also show that the proposed system can be combined with any other offline signature verification system to increase the number of samples on the enrollment set, leading to improved Equal Error Rate, including when used to detect skilled forgeries.

Resumo

Um dos principais desafios de sistemas de verificação de assinaturas *offline* 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 offline baseado em características dinâmicas. Em contraste com o estado-da-arte, o método proposto se baseia na perspectiva da aprendizagem supervisionada, o nossa máquina de aprendizagem é treinada para realizar a tarefa de "conversão de assinatura *online* para *offline*". O método proposto é uma *Deep Convolutional Neural Network* treinada para aprender como textos manuscritos *online* da base IRONOFF são transformados para o domínio *offline*. O objetivo principal do método proposto é o de aumentar sinteticamente bases de assinatura *offline* baseando-se em amostras *online* em direção a uma melhora nas taxas de reconhecimento de sistemas de verificação de assinaturas *offline*. Para isso, uma avaliação na base de assinaturas BiosecurID é realizada. Mostra-se que o modelo de rede profunda proposto consegue aprender como converter os dados biométricos *online* para criar a respectiva imagem em escala de cinza da assinatura. Especificamente, observa-se que as amostras sintéticas geradas com o método proposto são comparáveis a amostras reais, obtendo uma performance comparável ao de assinaturas reais. Também é mostrado que o sistema proposto pode ser combinado com qualquer sistema de verificação de assinatura *offline* para aumentar o número de amostras no conjunto de treinamento, levando a uma melhora no *EER*, inclusive quando usado para detectar forjas habilidosas.

Contents

List of Figures

List of Tables

List of symbols and abbreviations

AER	Average Error Rate
ANN	Artificial Neural Network
AHSVS	Automatic Handwritten Signature Verification System
CNN	Convolutional Neural Network
DET	Detection Error Tradeoff
DPI	Dots per Inch
EER	Equal Error Rate
FAR	False Acceptance Rate
FCN	Fully Convolutional Neural Network
FRR	False Rejection Rate
LReLU	Leaky Rectified Linear Unit
MSE	Mean Squared Error
MLP	Multilayer Perceptron
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine

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 is, so far, one of the primary methods for identity authentication: 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 variety 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 is a time-consuming, heavy-load and expansive task. Hence, several Automatic Handwritten Signature Verification Systems (AHSVS) have been proposed [2]. These systems aim to automatically decide if a query signature is genuine or not.

An AHSVS is essentially a pattern recognition system that receives a signature as input, extracts a set of features and classifies the sample using a template as the reference. Two acquisition modalities are available based on the specific application: signatures can be acquired by using an optical scanner or by using an acquisition device such as digitizing tablets or electronic pens with digital ink. The two domains are identified as offline (static) [4] and online (dynamic) [5], respectively. In the offline case, only a static representation of the completed writing is available as an image, while in the online modality data is stored during the writing process and consists of a temporal sequence of values. Each representation has, in general, specific attributes not present in the other [6].

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 [6].

Performance of an AHSVS also depends upon the amount of samples used for training. In order to increase the amount of training samples, synthetic signatures have been used for the online domain [7, 8] as well as for the offline one [9, 10].

Static signature images can be generated from online data [11, 12, 10, 13]. This approach can be used also to improve performance of an online signature verification system by coupling it with a parallel offline one [14].

However it must be argued that despite the advancements for this category of synthesis, the synthetic offline samples created up to date still struggle to improve the recognition rates when used to enlarge existing offline signature databases.

In this work, an approach based on Fully Convolutional Neural Networks (FCN) trained in a supervised manner is presented with the aim 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 since the neural network learns how the dynamic features, in particular pressure, of the online representation translates into the static domain (pen on the paper).

Differently from other models [9, 10, 11], the approach here presented is designed under the perspective of supervised training: a learning model is trained to

perform the task of "online to offline conversion" using (for training purpose) 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 so that synthetic signatures can be used to improve recognition rates).

In order to achieve this goal, the following steps have been performed: (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) comparison of the performance with a state-of-the-art classifier to evaluate the closeness of synthetic and real signatures. We also evaluate the use of synthetic signatures in the genuine enrollment set, analyzing the differences of the offline verification system recognition rates.

1.1 Problem Statement

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) Comparison of the proposed approach's performance with a state-of-the-art method 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 in the offline verification system recognition rates.

1.2 Dissertation Structure

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

Chapter 2 gives an overview on the topic of handwritten signature verification.

We introduce handwritten signatures as a biometric 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 Deep Learning research field. Chapter 4 describes our proposed method. Chapter 5 presents the experimental protocol followed in the experiments of this work and Chapter 6 presents and discuss the results. Finally, in Chapter 7 we present the conclusion of this study together with suggestions for future work.

C H A P T E R

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 analysis of biological or behavioral 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 measurable characteristic used to distinguish and describe individuals [15].

Biometric systems are often categorized as physiological or behavioral [16]. The physiological category is characterized by measurements of the body, such as fingerprint, face, DNA, iris/retina pattern, and body scent. On the other hand, behavioral biometrics are individual acquired traits and are related to the pattern of behavior of a person. They include typing rhythm, temperament, voice, and handwritten signatures [17].

Most biometric identifiers require a special type of device for security and control

of human identity. However, handwritten signature based biometric systems can be performed requiring no sensor except a pen and a piece of paper. According to [18] 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 [19].

A generic handwritten signature-based biometric system is shown in Figure 1. Once the user Y deposits the signature, a sensor digitalizes the sample. Later, a feature matrix X is built with the information extracted from the input sample. Next, the systems typically have two stages: enrollment X_E and recognition X_R. The former builds a system database 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 S is obtained according to the similarity of the questioned sample to the claimed template. Finally, the system accepts or rejects the questioned sample.
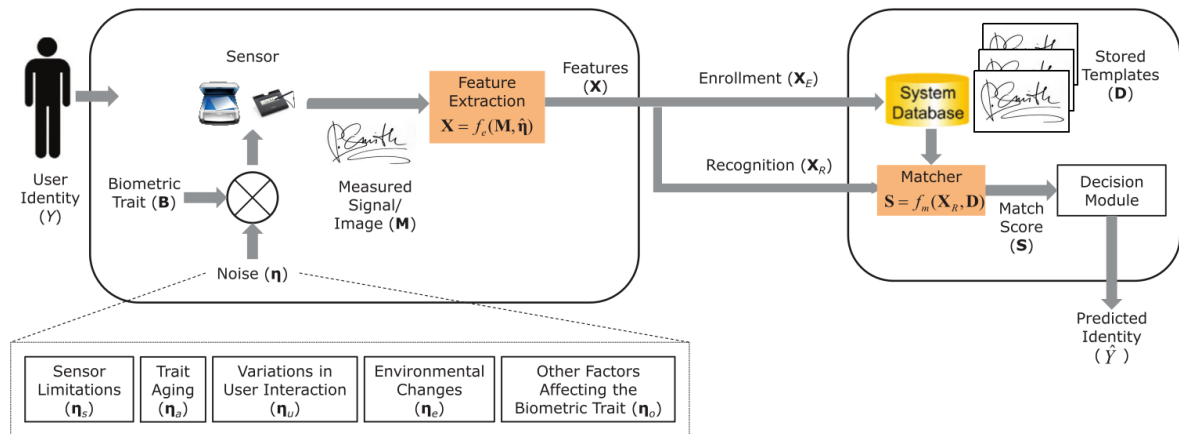
Figure 1 – Overview of a typical handwritten signature based system. Figure adapted from [17].

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 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 [20].

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 [11]. 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 [21]. In Figure 3 we can see a grayscale offline signatures example and the respective online signature plot.
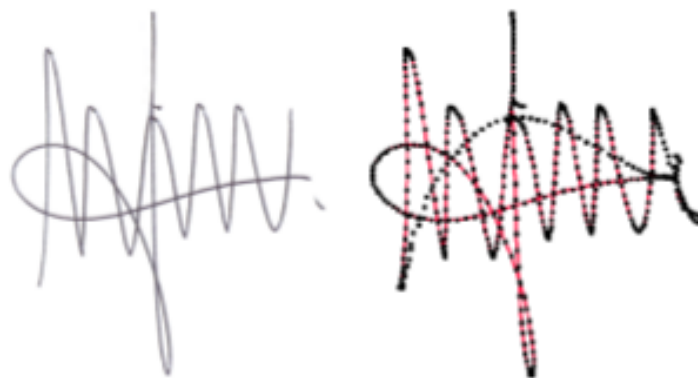
Figure 3 – An offline and a matching online signature sample. Figure extracted from [22].

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 and the paper [23]. According to [24] 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 (η) that distort the measured trait. According to [17] and shown in Figure 1, the intra-personal variability affecting the measured sample $\mathbf{M}$ 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 [4].

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.

In the field of signature verification, forgeries are usually classified into two types [2].

- 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 using his signature instead of the

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 [25].

signature to be tested. The forger does not attempt to simulate or trace a genuine signature.

- The second type is the skilled forgery, in which the forger tries and practices imitating as closely as possible the static and dynamic information of the genuine signature model. The forger has access to both the user's name and signature and tries to reproduce it with a similar intra-class variability.

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 [26].

As any Pattern Recognition system, an AHSVS has three phases: (i) data acquisition and pre-processing (ii) feature extraction (iii) 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 binary classification problem, in which the system's prediction to the input signature sample is either genuine or fraudulent.

Verification errors occurring in AHSVS are usually categorized as two types [27]. 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.

The performance of signature verification systems can be improved by increasing the number of samples in the training dataset. The amount of data available for each user is often insufficient in real applications. 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.

2.3 Off-line Signature Synthesis Using On-line Samples

According to [12], although online signature samples can be effectively stored as a time-series for use in any form of an AHSVS, there may be a situation where the signature might be needed to be represented as an image.

One possible scenario is for human visualization. For instance, in a context of a legal document, it may be required to have a image-based representation of signature that was captured in a biometric system and stored as a time-series. There may also be a case where the modality of the signature used for training the user model differs from the signature domain that is being questioned. Although the test sample is a genuine signature, it might not be possible to prove with either an offline or online signature verification system alone. Hence, converting either the training or questioned data into an image reproducing the original static signature would be useful for the development of an integrated version of an offline and online signature verification systems, overcoming the dynamic vs. static dichotomy, and unifying the signature biometry [14].

The works proposed in the literature to generate signature images from online signatures, generally apply different transforms to the dynamic information taking into account the kinematic and the timing order in which the traces were registered, then the samples of the new specimen are interpolated in order to create new images.

The accuracy of a series of interpolation methods for recreating a signature image from the online dynamic data was investigated in [12]. They experimented what they refer to linear interpolation, nearest neighbor interpolation, cubic spline interpolation, and Piecewise cubic Hermite interpolation. They experimented several interpolated line thickness widths, varying according to the pen pressure/force. In the work, the authors measured the performance of the synthetic samples using three statistics: the

mean Euclidean distance between the synthetic sample and the real, the maximum distance in pixels between the two images, and the percentage of pixels in the recreated image. They found that the linear technique with variable-width produced the best accuracy with 3.047, 67.147 and 27.416 mean pixel distance, percent direct match, and max distance, respectively.

In [10] another method to generate synthetic offline signature databases starting from synthetic dynamic data is described. They convert the dynamic information to offline data using what they refer to as an ink deposition model. In short, they first create an 8-connected sequences from the online signature, then use an approach to model the pen ballpoint following a 2D Gaussian function. In [11] they used this approach to generate synthetic samples based on real online data. They proposed an experimental protocol to assess whether the synthetic signatures are close to the real ones and to measure if it would be feasible to use these synthetic signatures to increase the training data of offline signature verification systems. They observed that the synthetic samples have a close performance to real signatures, but when used to increase the training dataset synthetically their synthetic samples achieved 18.70% EER, while the real samples 17.68%, in the skilled forgeries scenario. Moreover, in [13] they also experimented their synthesis approach to improve the performance of dynamic signature verifiers.

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 [28]. 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 [28].

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 [29]. 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.
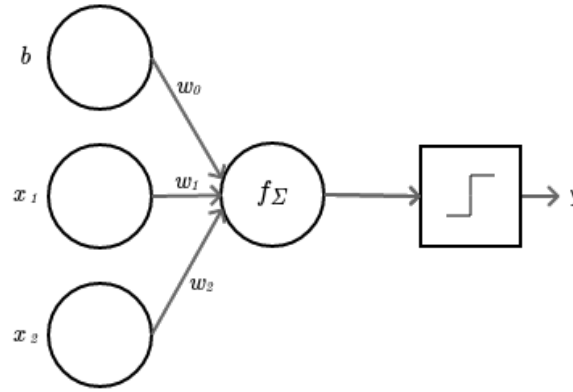
Figure 6 – Perceptron representation. x_1 and x_2 represent the input signal, b the bias term, w_0, w_1, w_2 the weights, f_Σ is the activation function (in this case, a step function) and the output signal is given by y.

Models based on perceptrons have severe limitations. An evaluation by Minsky and Papert [30] 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" [31]. 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 Multi-layer Perceptron (MLP) with a single hidden layer, however, has been proven [32] to approximate any continuous function on a compact input domain to arbitrary precision. For this reason, MLPs are said to be *universal function approximators* [32].

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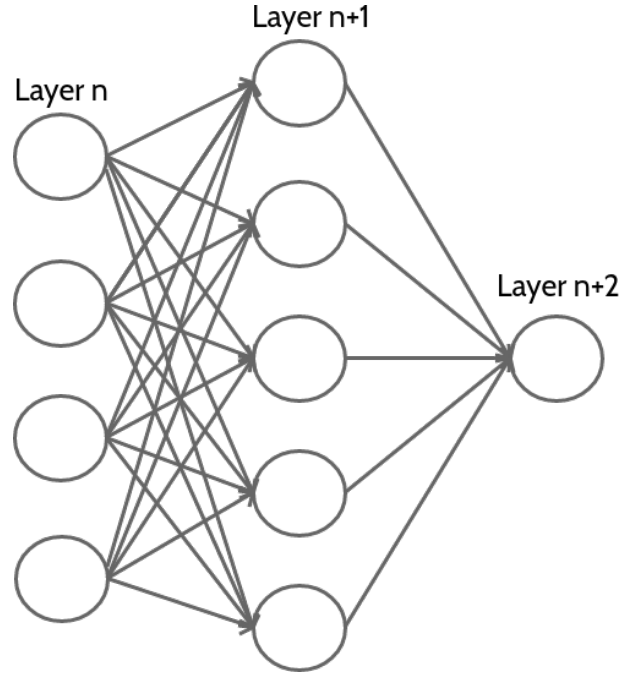
Figure 7 – Multilayer perceptron representation. Each layer contains several perceptron units, which are then connected to units in the subsequent layer.

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 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 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 a 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). Then, it propagates 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 the 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 [33]. The parameter α is a heuristic, called learning rate. The learning rate values help to avoid convergence to a local-minimum or maximum point.

3.2.3 Activation Functions

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 and is shown in the Figure 8 (a).

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

It takes a real-valued number and transforms it to the range between 0 and 1. Small negative numbers become 0 and large positive numbers become 1.

(a) Sigmoid

(b) Tanh

(c) ReLU

(d) LReLU
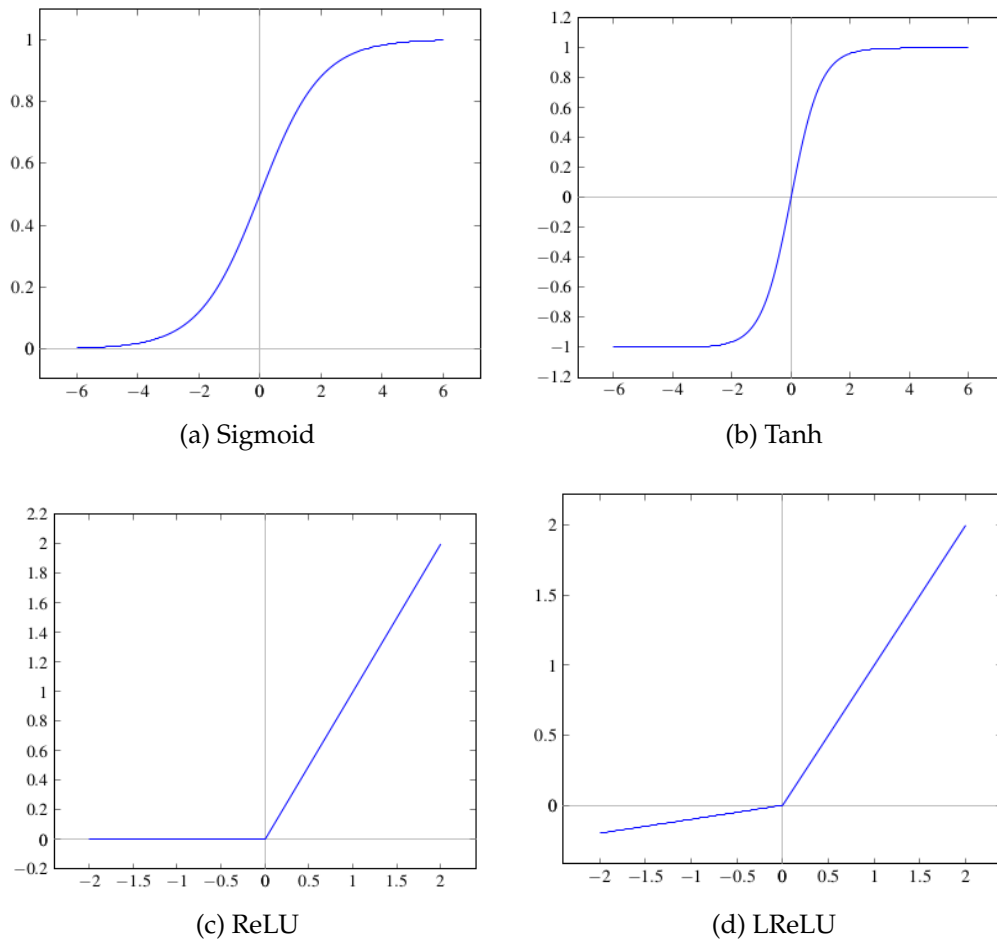
Figure 8 – Neural network activation functions.

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 Rectified Linear Unit

The Rectified Linear Unit (ReLU) has become very popular in the last few years and was one of the main discoveries that enabled the training of deeper neural networks. It computes the function

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

this activation function can be simply regarded as a threshold at zero, see Figure 8 (c).

3.2.3.4 Leaky Rectified Linear Units

Leaky Rectified Linear Units (LReLU) is an attempt to improve the ReLU. Instead of the function being zero when x 0, a leaky ReLU will instead have a small negative

slope (around 0.01). The function computes

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

where α is a small constant, see 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). It was found, however, that the mammal brain is organized in a deep architecture. The brain appears to process information through multiple stages, which is particularly clear in the primate visual system [34].

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 [35] 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 [36]. Besides, the theoretical findings include unsupervised training of the layers [37], Rectifier Linear Unit (ReLU) [38] as the activation function and the regularization techniques Dropout [39] and Batch Normalization [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 achieves state-of-the-art performance on visual object recognition and image classification tasks [36]. 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 an architecture of a CNN, namely, the Lenet [41], one of the first applications of the CNN used for handwriting recognition.
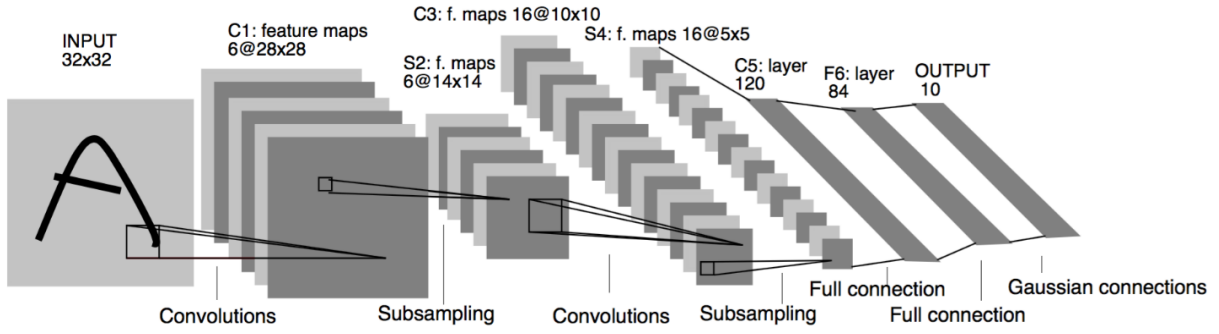
Figure 9 – Lenet [41], a convolutional neural network used for handwriting recognition.

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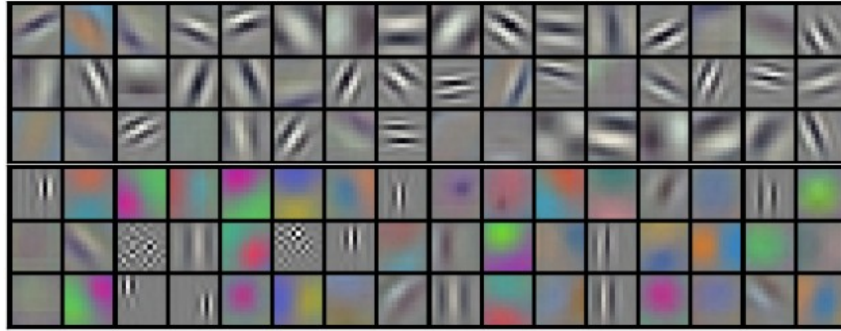
Figure 10 – Example of feature maps learned by the CNN proposed by [42] for object classification.

3.4.1 Convolutional Layer

Convolutional layers have trainable filters (also called feature maps) that are applied across the entire input [35]. 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 an 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).
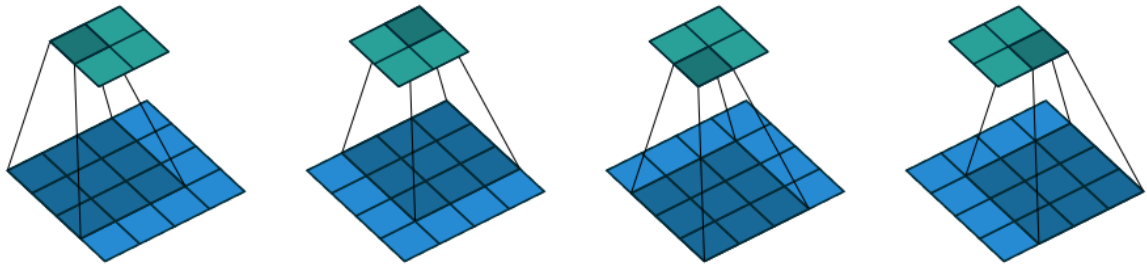
(a) Convolving a 3x3 kernel over a 4x4 input, with 1x1 stride.
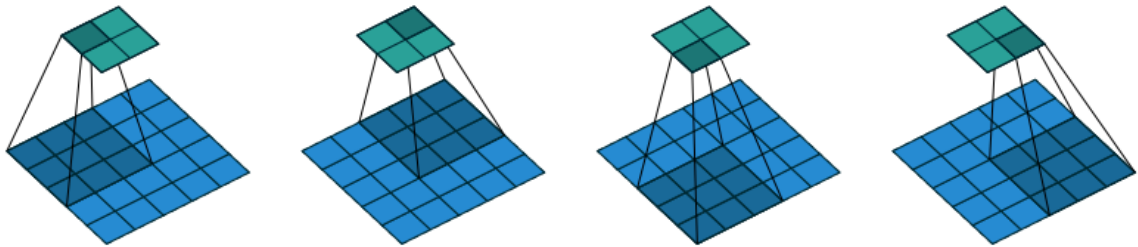
(b) Convolving a 3x3 kernel over a 5x5 input, with 2x2 stride.

Figure 11 – Convolution operation with (a) stride 1x1 and (b) stride 2x2. 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).

---

[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>

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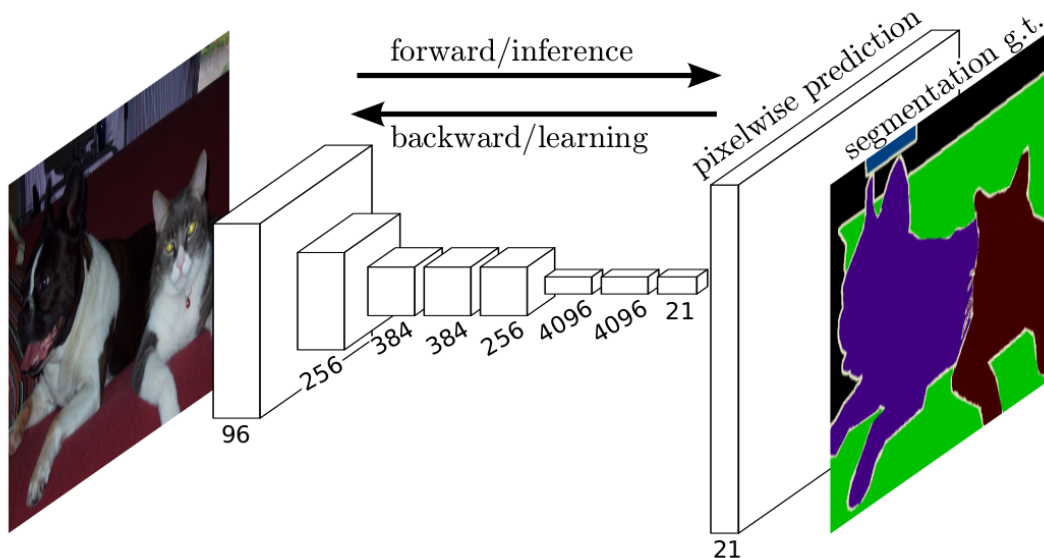
Figure 12 – Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks. Extracted from [44]

3.5.1   Transposed Convolution

Transposed convolutions - also called deconvolution or backward convolution - work by swapping the forward and backward passes of a convolution. Whether it is a direct convolution or a transposed convolution is determined by how the forward

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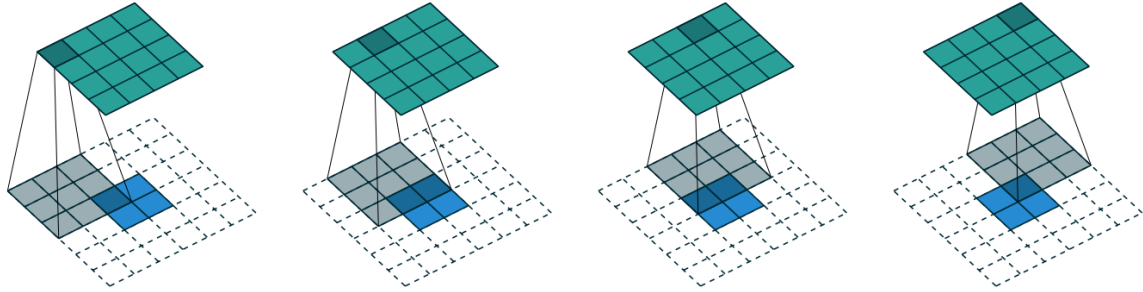
Figure 13 – Upsampling with transposed convolutions. Upsampling is done by padding (white) the original pixels (blue) and convolving them with a filter (gray). The result is an upsampled image (green). Here a stride of one is used, illustrated from left to right. Extracted from [43].

3.6 Deep 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 the neural network converges faster using the weight initialization described in Equation 3.11.

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

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 Algorithms

Training the network consists in minimizing the error function, based on the gradients of the parameters with relation to the cost function, by updating the weights and biases, expecting that the network learns how to perform the task it is being trained for.

3.6.2.1 Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) algorithm, as defined in [31] can be summarized a series of iterations 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

The Adaptive Moment Estimation (ADAM) [46] is a stochastic optimization method that can adaptively tune the learning rate per parameter and also keeps an exponentially decaying average of past gradients m_t and v_t:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.12a}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.12b}$$

where m_t and v_t are estimates of the first moment and the second moment of the gradients respectively, β_1 and β_2 are the decay rate and g is the gradient with respect to w.

Since the moments are biased towards zero mostly during the initial time steps, the authors work around these biases by computing bias-corrected estimates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.13a}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.13b}$$

Then, they use these computed variables to update the parameters:

$$w_{t+1} = w_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.14}$$

The authors propose default values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ϵ.

4

Proposed Method

Given a dynamic signature, the simplest way to obtain the corresponding static sample is to draw (x,y) coordinates and performing a simple interpolation in order to get a continuous trajectory. The problem of this approach is that other dynamic information (e.g. velocity, pressure, etc.) do not contribute in the transformation process, so that the offline version will result in a poor execution.

Some more complex methods have already been proposed, nevertheless, these methods are "rule-based". Here we propose a method that learns from data how to perform the transformation having samples of both dynamic and static signatures.

Figure 14 shows the workflow of the proposed approach to generate a static signature-based on online data. In this chapter, we describe our proposed method. First, the dataset used to train our model is described, then we describe how we preprocessed the dataset. Afterwards, the FCN model trained to learn an end-to-end mapping from the dynamic information to the static representation of the signature is detailed.

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 manuscript containing the trajectory and pressure information mapped to the respective resulting offline representation, as in Figure 3. Of course the mapping is a major problem since the two acquisition systems (digitizing tablet and optical scanner) process the writing signal separately, each having its particular parameters (Cartesian axis, scale, etc.).
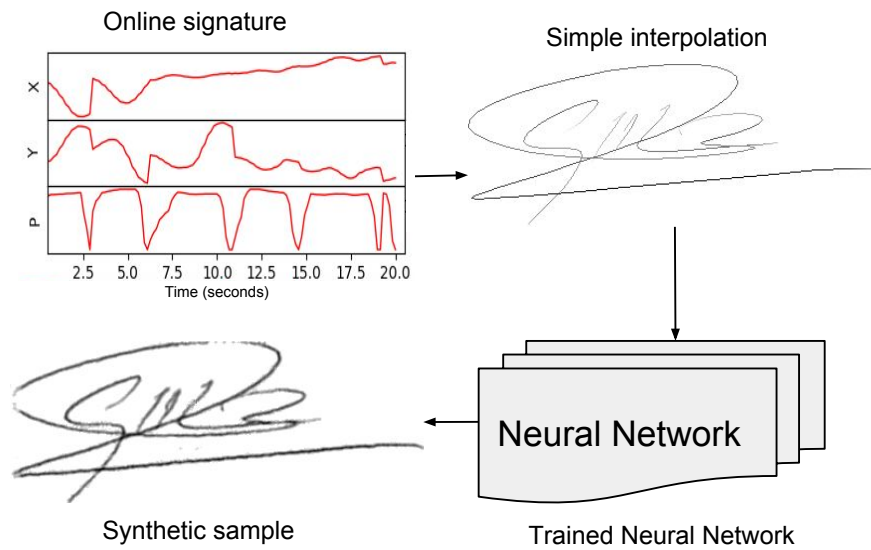
Figure 14 – The proposed approach diagram and an example of the synthetic signature generation.

Moreover, the acquisition phase of such dual dataset requires some extra steps. A paper form must be placed over a digital tablet to be filled with an electronic ink-pen. Then, the dynamic data is captured through the tablet, and the paper form might be scanned to provide the offline image, Figure 15 illustrates this setup. Thereby, two complementary files are available. One contains the dynamic information, and the other one is the digital image of this piece of handwritten signature produced by the scanner.

In order to project the online signature in the respective offline version, these two types of information should be available within the same coordinate system, with the same origin, the same resolution, and orientation. However, as the two acquisition systems (tablet and scanner) are processing the writing separately and with their particular parameters, this assumption is not satisfied directly. Geometrical transformations have to be applied to compensate for these differences. To the best of our knowledge, however, the creators of the publicly available dual modal signature datasets [47, 48, 49, 22, 50, 51, 52] had not this characteristic satisfied. As it can be seen in Figure 16, both of the representations of the signatures do not match if we plot it in a single image.

The dual domain IRONOFF [6] handwriting dataset was thus used to train our model. Besides acquiring both domains of the handwriting manuscript, the online data is mapped to the same coordinate system of the offline data, as shown in Figure 17, the interpolated online data is shown in white. Using this dataset, we make the fair assumption that a handwritten signature is a manuscript. With that in mind, we expect that even if the network was trained on handwriting manuscripts, it would work for online signatures to generate its static version.

Figure 15 – An user signing on a paper placed over a digitizing tablet. This way, the dynamic and static versions of the same signature are acquired simultaneously. Figure extracted from [13]
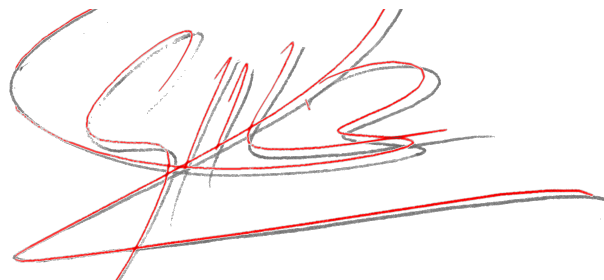
Figure 16 – A sample from the BiosecurID dataset. Here we can see that the online signature (interpolated in red) can not be projected in the respective offline version.

The IRONOFF dataset contains a total of 23000 mapped online and offline samples of the 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).

Figure 17 – An offline manuscript mapped with the respective online trajectory. Image
extracted from [6].

4.2   Preprocessing

During the training of our FCN model, two 2D matrices are needed, the input
and output. The input is the online information of the manuscript, and the output is the
expected offline representation.

In order to create a 2D representation of the dynamic information of the manuscript,
we create a simple linear interpolated version of the time-series (x_t, y_t, p_t) using the
Bresenham line algorithm [53] to obtain the 8-connected sequences. As a result, we get
an image in which the trajectory of the signature is represented by one pixel, where the
pixel intensity is the pressure information.

Since the neural network expect inputs and outputs of a fixed size and the
manuscripts shape vary significantly in the IRONOFF (they range from samples of
width x height size of 167x214 to larger samples of size 548x215 pixels), we convert it
to a fixed size. We follow an approach similar to what was previously proposed in the
literature, e.g., [54]. First, we normalize the images to the largest image size, by padding
the images with white background. Then, we center the manuscript in a canvas of size
548x215 pixels (the largest sample size in the dataset), aligning the center of mass of
the sample to the center of the image. Finally, we rescale the images, using a bilinear
interpolation, 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.

Figure 18 shows what is the result of the preprocessed input and the groundtruth

being fed to the neural network during the training phase.

(a) input   (b) ground truth

Figure 18 – 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.

4.3   Neural Network Model

Our model is based on the fully convolutional neural network architecture, adopted to learn an end-to-end nonlinear mapping from online representation to the static image. 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.

We use a simplified architecture based on the one proposed in [44]. 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 on the online sample.

Our convolutions architecture is a simplified version of the FCN-VGG [44, 55] and the transposed convolutions were symmetric to the convolution operations. We used a simplified version because during some initial tests we observed that the capacity of the original network architecture seemed to be too large for the problem at hand, particularly considering the amount of convolutional operations. We observed that the model was not learning useful transformations on the first few epochs. We obtained better results with the simplified architecture presented in Table 1.

For the purpose of replicating our experiment, we provide the list of the parameters used in our tests. Table 1 lists the definition of the FCN 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 are the width of the layer window.

We used Leaky Rectified Linear Units (LReLUs) as the activation function for all convolutional layers. We also tried the Rectified Linear Units (ReLUs) but, although not performing extensive tests, we observed that on the first ten epochs the Neural Network seemed to be converging faster with the LReLUs.

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

For every convolutional layer we use a stride (the distance between applications of the convolution operation) of 2x2, following what was observed by [56] that we can remove each pooling layer and increase the stride of the convolutional layer that preceded it accordingly. We pad the input for every layer with 0 evenly left and right and we initialize the weights of the model using the technique proposed by Glorot *et al*[45], and the biases to 0.

We trained the model using the 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. We used 22000 samples from the IRONOFF to train the model and another 1000 samples as validation data. The network was trained using the library Tensorflow and took around five days to train on a GTX 670 GPU.

4.4   Training Results

The MSE loss on the validation set over the 100 training epochs can be seen in Figure 19. We can notice that the MSE loss starts at 0.007 on the first epoch, and decreases until epoch 20 where it starts varying from around 0.002 to 0.001 until the last epoch.

A visual representation of the last epoch model predictions, alongside the expected output is shown in Figure 20.

In order to make an objective evaluation of the quality of the synthetic generation, in the context of handwritten signatures, we performed a machine-oriented evaluation. In the next chapter, we describe the protocol we followed and then we report an objective measure of the quality of our proposed method based on this experimental setup.
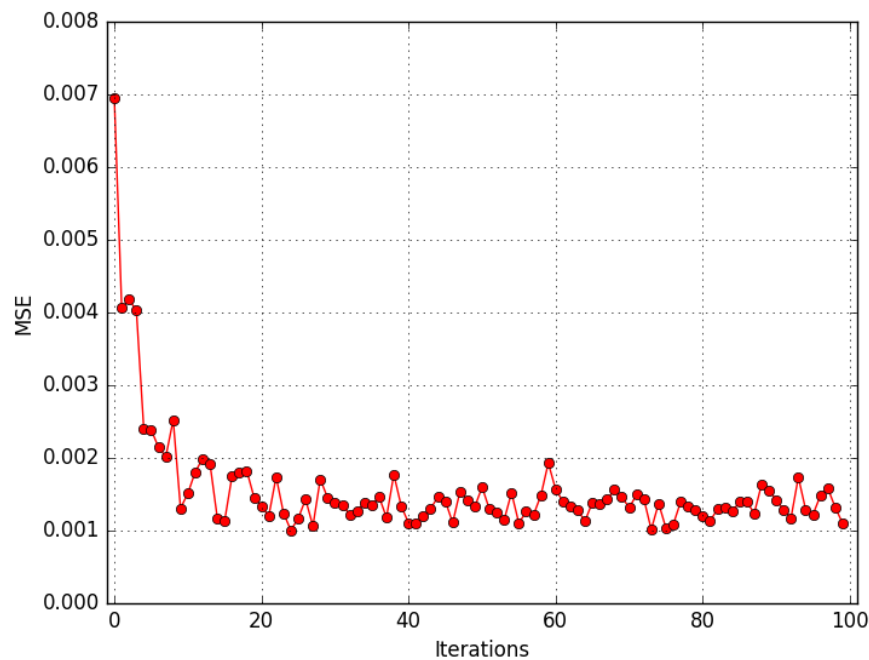
Figure 19 – Iterations versus validation loss.

(a) Synthetic Handwritten Manuscript

(b) Expected Real Manuscript - Groundtruth
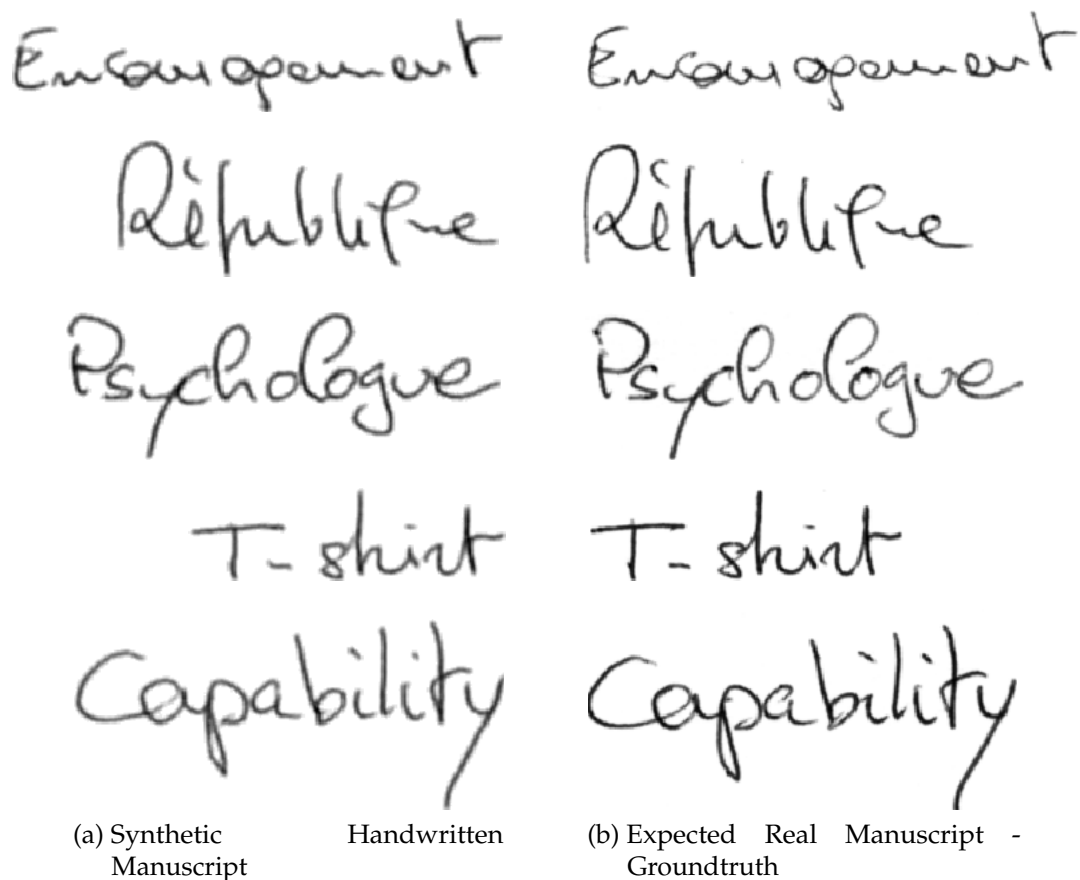
Figure 20 – A not cherry-picked selection of synthetic manuscripts produced using our proposed method alongside the expected output, i.e. the groundtruth.

CHAPTER 5

Evaluation Setup

The quality evaluation of the synthetic signatures has been performed according to the protocol presented by Diaz-Cabrera et al. [11]. A state-of-the-art offline verification system and a dataset including both online and real offline signatures was considered in order to train the system. The main idea of experiments is to measure the quality of the synthetic signatures in terms of performance comparison to the use of only real offline signatures and in inspecting performance variations by increasing the number of training samples with synthetic generated ones.

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 offered by real offline signatures? (ii) is it feasible to increase the number of samples at the enrollment stage with our proposed method synthetic signatures?

5.1   Offline Signature Verification System

The system used for the evaluation of the real and synthetic signatures is a Linear SVM classifier and with a state-of-the-art feature extraction approach [57]. The feature extraction system [4] uses ideas from transfer learning and multi-task learning to learn features using Convolutional Neural Networks (CNN). As discussed in Chapter 3, one of the advantages of using deep learning techniques is that some models, such as the CNN, can learn filters that can be used as feature extractors. The offline system's feature

---

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

extractor takes advantage of this concept.

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 plane such that it maximally separates the margin between the two classes, as it can be seen in Figure 21. For this reason, SVMs are also referred to as maximum margin classifiers.

(a) The data can be separated using different. lines

(b) The SVM strives for the optimal line, which maximally separates the margin between the two classes of data.
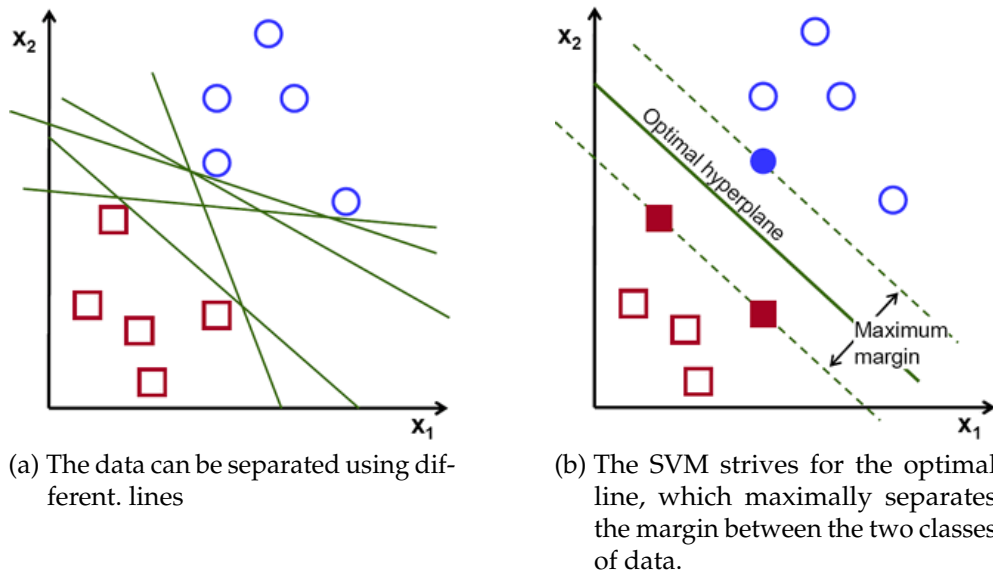
Figure 21 – Illustration of several separating lines and an optimal line.

5.2 Evaluation Database

The evaluation experiments were carried out on the BiosecurID database [47]. This multimodal database was made publicly available containing signatures of 132 subjects. Since the signatures were captured using a special digital inking pen on a paper placed over a digitizing tablet, as shown in Figure 16, the online and respective offline version of the same real signature were acquired simultaneously. This characteristic, therefore, makes BiosecurID the ideal benchmark for the experimental evaluation conducted in this work.

The signatures samples were captured in 4 different sessions, distributed over four 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.

Since the offline signature verification system is trained with both positive and negative samples, to ensure an unbiased result towards the dataset selection, similar to

the protocol followed by [11] we used the MCYT database [25] as the negative dataset samples. The MCYT dataset includes 75 signers each with 15 genuine and 15 skilled forged signatures. The amount of negative samples was set to 25 random samples from the MCYT dataset.

5.3 Experimental Protocol

In order to accomplish a fair comparison of our work and the state of the art, the same experiment protocol proposed by [11] has been adopted. More specifically, two different experiments are carried out: Experiment 1 focuses on evaluating the synthetic signatures performance in comparison to real signatures; 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. According to [11], the first 90 users are used as the enrollment set in order to compute the genuine and skilled impostor scores. The remaining 42 users 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. We report the details of experiments setup for an easy reading of the paper.

- Experiment 1:Two different protocols in order to train the system are considered: (i) mono-session: four samples of the first acquisition session are used; (ii) multi--session: one sample of each of the 4 acquisition sessions are used. Genuine scores are evaluated using the user's genuine samples not used in enrollment. Random forgery scores are evaluated by comparing the first sample of the test subjects to the enrolled model. Skilled forgery scores are evaluated with the skilled forgeries samples of the enrolled users.

- Experiment 2: This experiment is designed to assess whether synthetically increasing the enrollment dataset leads to a better recognition performance. The evaluation is made through three different enrollment sets:

  - 4 real samples from the first acquisition session
  - 4 real samples from the first plus 4 real samples from the second session
  - 4 real samples from the first session plus 4 synthetic samples of the second session.
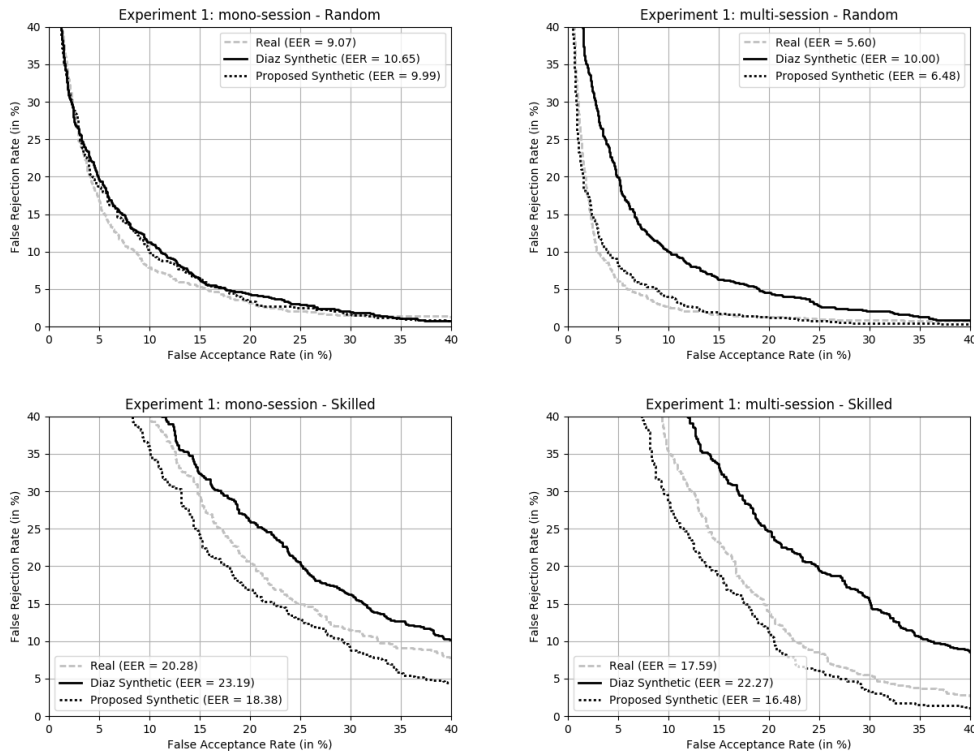
Figure 22 – DET curves for real offline signatures and synthetic signatures (from Diaz and our proposed method), for the first experiment (mono-session and multi-session enrollment), for the two scenarios considered (random and skilled impostors)

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. In our experimental results, the system's EER performance is reported.

CHAPTER

6

Results

As described in Chapter 5, two different experiments were carried out to evaluate the synthetic signatures generated with our proposed method. Namely, we intend to measure how close the synthetic signatures are to real ones and to verify the feasibility of increasing the enrollment dataset with synthetic signatures.

The goal of the experiments are:

- Measure the quality of the synthetic signatures samples in terms of similarity to real specimens;

- Evaluate if using synthetic signatures affects an offline signature verification system performance; and

- Analyze whether using real or our proposed method synthetic signatures on the enrollment set have similar recognition rates.

We compare our results with a state-of-the-art method, particularly the approach proposed by Diaz *et al.* [11]. Specifically, our proposed method is compared with the "Image enhanced" synthetic signatures made available as part of the BiosecurID [47] dataset. The reported EER is achieved for both approaches in 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 *et al.* [11]; (iii) synthetic signatures created with our proposed method.

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 scenarios were experimented:

- Mono-session: signatures from the first session on the enrollment set; and

- Multi-session: one signature per session.

Table 2 – EER for real, synthetic samples from Diaz *et al.* [11] 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 *et al.*	Proposed method
mono-session	20.28%	23.19%	18.38%
multi-session	17.59%	22.27%	16.48%
	Random Forgeries		
	Real	Diaz *et al.*	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. We observe that under the skilled forgeries scenario our proposed method synthetic signatures EERs yields better performance than both the real dataset and the synthetic signatures generated by Diaz *et al.*. On the random forgeries test, we can identify that, under the random forgeries scenario, the EER results achieved by the real, and both types of synthetic signatures are close to each other. Specifically, under the mono-session scenario, the system's performance for the real signatures was 9.07%, while the synthetic samples from Diaz *et al.* and our proposed method achieved 10.65% and 9.99%, respectively. However, on the multi-session test, the signatures from Diaz *et al.* achieved 10.00% ERR, while the real signatures and our proposed method EER was 5.60% and 6.48%, respectively.

In Figure 23 (left - mono-session, right - multi-session) we may observe that the behavior of the system with real (gray dotted) and our proposed method synthetic signatures are quite similar, regardless of the training protocol or the scenario considered.

Nevertheless, our proposed method synthetic samples achieved a better discriminative power for skilled forgeries.

Figure 23 – DET curves for real offline signatures and synthetic signatures (from Diaz *et al.* and our proposed method), for the first experiment (mono-session and multi-session enrollment), for the two scenarios considered (random and skilled impostors)

Moreover, in Figure 24 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 is able to produce synthetic signatures comparable to the real signatures.

6.2 Feasibility of Synthetically Increasing the Enrollment Samples - Experiment 2

Figure 25 reports DET curves related to the second experiment, results are summarized in Table 3. EER decreases from 10.26% to 9.74% on the random forgeries scenario and from 21.55% to 19.17% for skilled forgeries, respectively. We can state

(a) mono-session - random forgeries

(b) mono-session - skilled forgeries

(c) multi-session - random forgeries

(d) multi-session - skilled forgeries
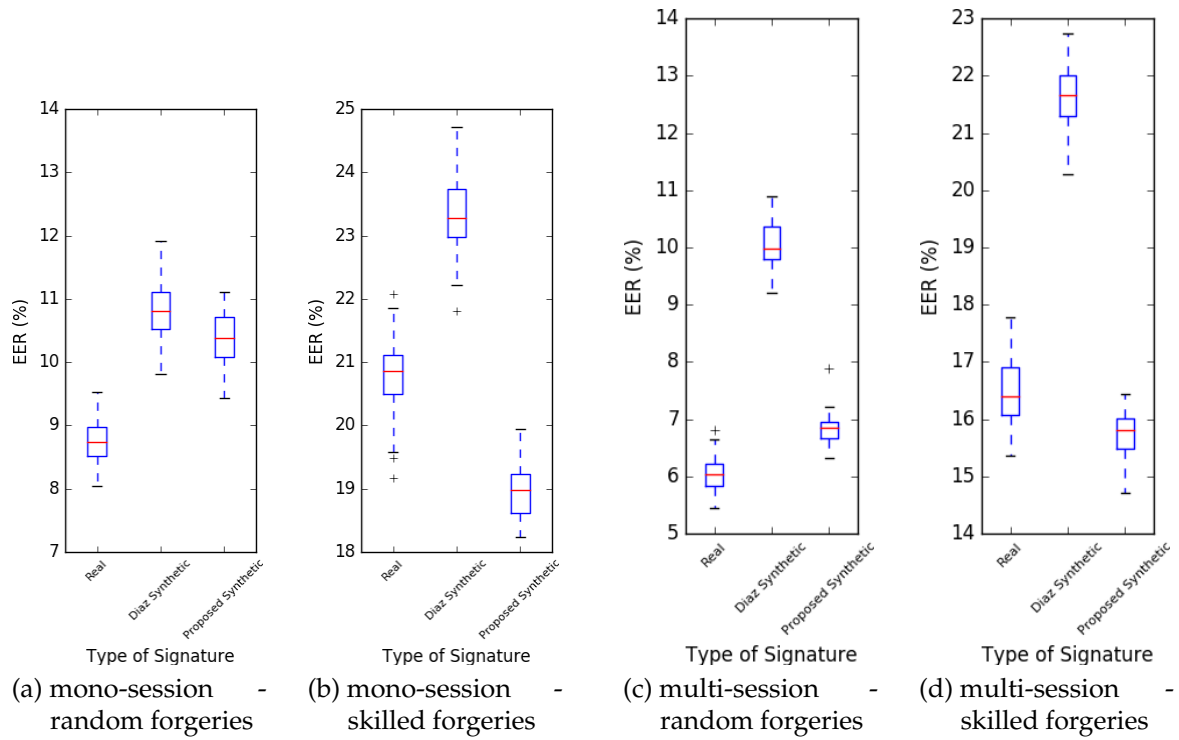
Figure 24 – Boxplot comparison for running 30 times the Experiment 1 - mono-session scenario and multi-session. (a) mono-session scenario, random forgeries, (b) mono-session scenario, skilled forgeries, (c) multi-session scenario, random forgeries, (d) multi-session scenario, skilled forgeries.

that the use synthetic samples for increasing the training set leads to better recognition performance.

Table 3 – EER for real, synthetic samples from Diaz *et al.* [11] 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 *et al.*	24.19%	7.72%
4 real + 4 synthetic from the Proposed Method	19.17%	9.74%

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.

Furthermore, we can see in Figure 26 that when running the experiments for 30 times, the results of our proposed method has a consistent mean, with a lower variance
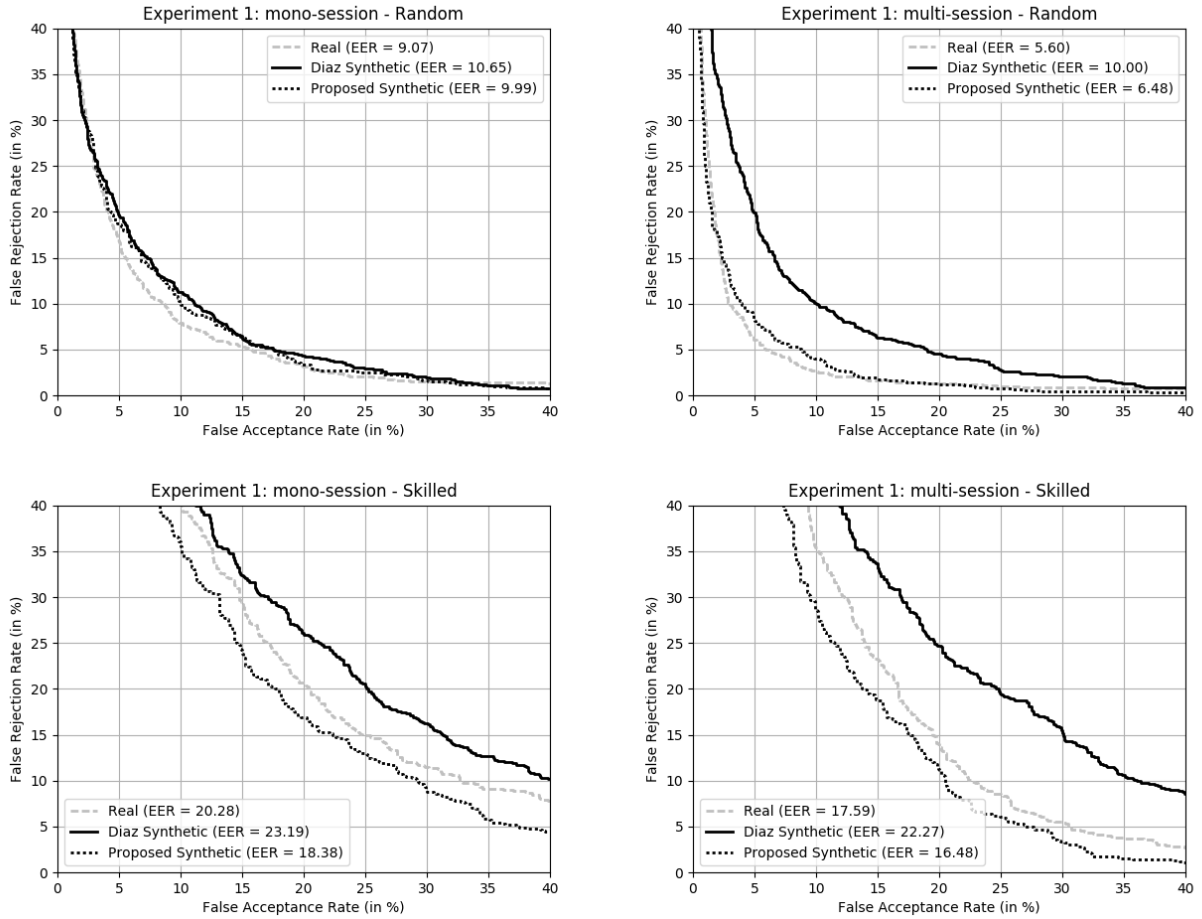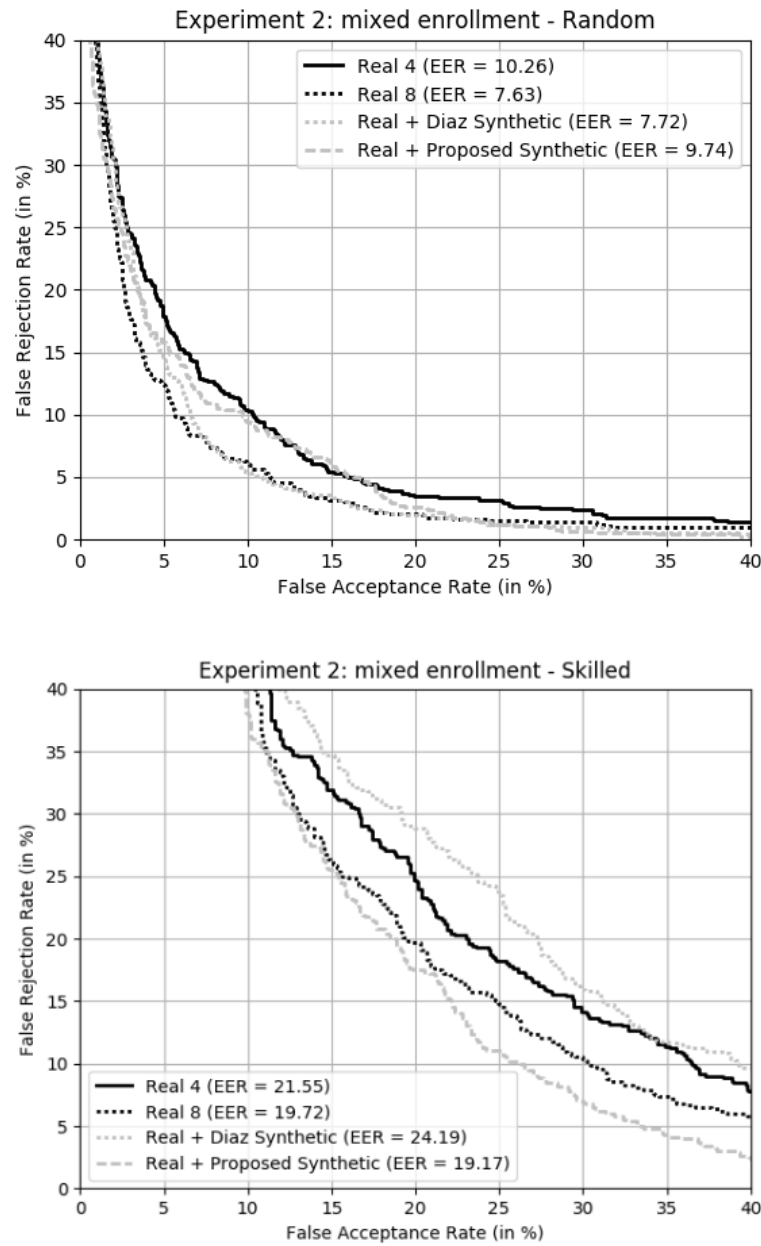
Figure 25 – DET curves for real offline signatures and synthetic signatures (from Diaz *et al.* and our proposed method), for the second experiment, for the two scenarios considered (random and skilled impostors)

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.

It is possible for us to conclude that it is feasible to use our proposed system to generate synthetic signatures from dynamic data, when complementary online samples are available, to increase an offline signature dataset, aiming to improve the recognition rate of offline verification systems.
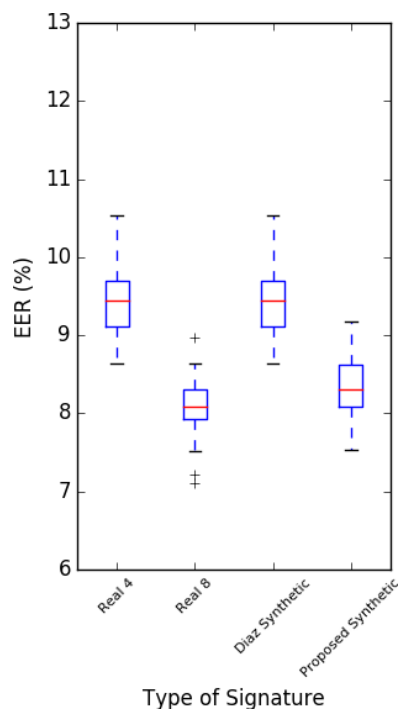
(a) mixed enrollment - random forgeries
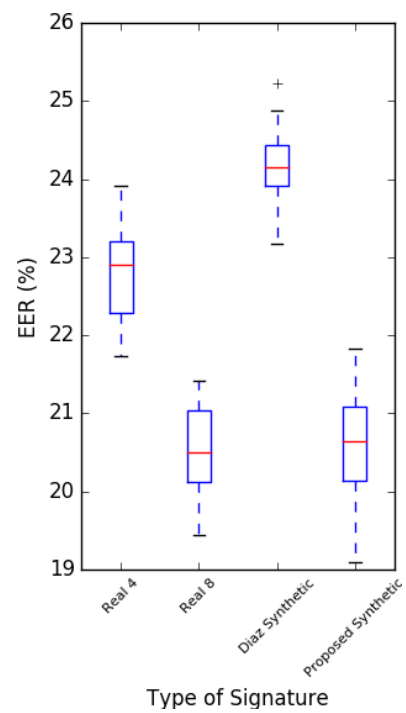
(b) mixed enrollment - skilled forgeries

Figure 26 – Boxplot comparison for Experiment 2: mixed enrollment - (a) random forgeries and (b) skilled forgeries.

CHAPTER

7

Conclusion and Future Works

In this work a method to produce synthetic offline signatures by means of a supervised learning task has been presented. We propose a Fully Convolutional Neural Network to learn an end-to-end mapping of online manuscripts to the static domain, taking advantage of the data present in the IRONOFF dataset during the training phase of the Neural Network.

We show that, in contrast to what has already been proposed, it is possible to model the "online to offline signature" as a learning from data task. Our Deep Neural Network model learns how the online biometric data must be used to build the grayscale signature image. The use of synthetically generated signatures offers verification performance similar to those observed by using real signatures. We also show that the synthetic signatures generated with our proposed method present high discriminative power when used to increase the enrollment set under the skilled forgeries scenario.

In order to evaluate our proposed method, we evaluated our proposed method by means of machine-oriented validations. We used a state-of-the-art automatic signature verifier to assess that similar results were obtained with both the real and our proposed method synthetic signatures. We found that our proposed method produces synthetic signatures comparable to real ones and when used to increase amount of training samples the system's EER decreases from 10.26% to 9.74% on the random forgeries scenario and from 21.55% to 19.17% for skilled forgeries.

Future work can explore the optimization of the hyper-parameters of the FCN (such as the number of layers, number of neurons per layer) since these hyper-parameters were fixed during the experiments in the present work. Besides, we intend to explore

the combination of real online and synthetically generated offline signatures using the proposed method, when only the online information is available, towards improved recognition results on a dynamic signature verifier.

It is also possible to explore recent advancements on Generative Adversarial Networks [58] such as the Image-to-image translation [59] method to convert the online samples to static images.

Another improvement to the current work is the integration of other dynamic features in addition to the pressure on the input of the network, such as the velocity. Moreover, one could also design a model to synthesize offline signatures with bigger resolution.

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

[5]   CPAŁKA, K.; ZALASIŃSKI, M. On-line signature verification using vertical signature partitioning. *Expert Systems with Applications*, Elsevier, v. 41, n. 9, p. 4170–4180, 2014.

[6]   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.

[7]   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.

[8]   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.

[9]   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.

[10]   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.

[11]   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.

[12]   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.

[13]   GALBALLY, J.; DIAZ-CABRERA, M.; FERRER, M. A.; GOMEZ-BARRERO, M.; MORALES, A.; FIERREZ, J. On-line signature recognition through the combination of real dynamic data and synthetically generated static data. *Pattern Recognition*, Elsevier, v. 48, n. 9, p. 2921–2934, 2015.

[14]   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.

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

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

[17]   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.

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

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

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

[21]   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.

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

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

[24]   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.

[25]   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.

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

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

[28]   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.

[29]   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.

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

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

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

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

[34]   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.

[35]   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.

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

[37]   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.

[38]   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.

[39]   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.

[40]   IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2015. p. 448–456.

[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] KINGMA, D.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54]   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.

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

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

[57]   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.

[58]   GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 2672–2680.

[59]   ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

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.