Offline Handwritten Signature Verification - Literature Review

Luiz G. Hafemann, Robert Sabourin
Lab. d'imagerie, de vision et d'intelligence artificielle
École de technologie supérieure
Université du Québec, Montreal, Canada
lghafemann@livia.etsmtl.ca, robert.sabourin@etsmtl.ca

Luiz S. Oliveira
Department of Informatics
Federal University of Paraná
Curitiba, PR, Brazil
lesoliveira@inf.ufpr.br

Abstract

The area of Handwritten Signature Verification has been broadly researched in the last decades and still remains as an open research problem. This report focuses on offline signature verification, characterized by the usage of static (scanned) images of signatures, where the objective is to discriminate if a given signature is genuine (produced by the claimed individual), or a forgery (produced by an impostor). We present an overview of how the problem has been handled by several researchers in the past few decades and the recent advancements in the field.

1 Introduction

Biometrics technology is used nowadays in a wide variety of security applications. The aim of such systems is to recognize a person based on physiological or behavioral traits. In the first case, the recognition is based on measurements of biological traits, such as the fingerprint, face, iris, etc. The later case is concerned with behavioral traits such as voice and the handwritten signature [1].

Biometric systems can perform two tasks: verification and identification. In the first case, a user of the system claims to be a particular person, and provides the biometric sample. The role of the verification system is to check if the user is indeed who he or she claims to be. In the identification case, a user of the system provides a biometric sample, and the objective is to identify, among all the people enrolled in the system, who the person is.

The handwritten signature is a particularly important type of biometric trait, mainly due to its ubiquitous use to verify a person's identity in legal, financial and administrative areas. One of the reasons for its widespread use is that the process to collect handwritten signatures is non-invasive, and people are familiar with the use of signatures in their daily life [2].

Signature verification systems aim to automatically discriminate if the biometric sample is indeed of a particular person or not, that is, they are used to classify query signatures as genuine or forgeries. This type of system usually consist of an enrolment phase, where the users of the system provide samples of their signatures, and an operation (or classification) phase, where a user claims the identity of a person and provide a query signature. The system then classifies such query as genuine (confirming the user identity), or as a forgery.

In the research of automated signature verification systems, forgeries are often classified in three types: random, simple and skilled (or simulated) forgeries. In the case of random forgeries, the forger has no information about the user or his signature and uses his own signature instead. In this case, the forgery contains a different semantic meaning than the genuine signatures from the user, presenting a very different overall shape. In the case of simple forgeries, the forger has knowledge of the user's name, but not about the user's signature. In this case, the forgery may present more similarities to the genuine signature, in particular for users that sign with their full name, or part

of it. In skilled forgeries, the forger has access for both the user's name and signature, and often practices imitating the user's signature. This result in forgeries that have higher resemblance to the genuine signature, and therefore are harder to detect.

Depending on the acquisition method, signature verification is divided in two categories: online (dynamic) and offline (static). In the online case, an acquisition device (such as a digitizing table) is used to acquire the user's signature. The data is collected as a sequence $S(n), n = 1, ..., N$, where $S(n)$ is the signal sampled at time $n\Delta_t$, Δ_t being the sampling interval. The signal may contain only the position of the pen, or include additional information such as the pen inclination, pressure, etc. In offline signature verification, the signature is acquired for the system after the writing process is completed. In this case, the signature is represented as a digital image, usually in grayscale format, as a set of points $S(x, y), 0 <= x <= H, 0 <= y <= W$, where H and W denote the image height and width [3].

Over the last few decades, some key survey papers have summarized the advancements in the field, in the late 80's [4], 90's [5] and 2000's [3]. This report reviews the most important techniques and recent advancements on the field of Offline Signature verification. It is organized as follows: we start by formalizing the problem at hand, and list the datasets that are available to evaluate such systems. We then describe the techniques used for each process of the pipeline for training a system: Preprocessing, Feature Extraction and model training, and finally we summarize the recent progress.

2 Problem Statement

In the literature of Offline Signature Verification, we can find multiple ways of defining the problem. In particular, one matter is critical to be able to compare related work: whether or not skilled forgeries are used for training. Some authors do not use skilled forgeries at all for training (e.g. [6], [7]), other researchers use skilled forgeries for training writer-independent classifiers, testing these classifiers in a separate set of users (e.g. [8]); lastly, some papers use skilled forgeries for training writer-dependent classifiers, and test these classifiers in a separate set of genuine signatures and forgeries from the same set of users.

In this report, we are concerned with the design of an Offline Signature Verification System that could be used in practice. For this reason, we restrict our analysis to methods that do not rely on skilled forgeries for the users enrolled in the system, since this is not the case in practical applications. We do consider, however, that a dataset consisting of genuine signatures and forgeries is available for training writer-independent classifiers, where the users from this dataset are not used for evaluating the performance of the classifier.

We first consider a development set $\mathcal{D}$ containing samples from a set of users $\mathcal{Y}_\mathcal{D}$. The development dataset is composed of genuine and skilled forgeries for the users in $\mathcal{Y}_\mathcal{D}$: $\mathcal{D} = \{\mathcal{D}_{genuine} \bigcup \mathcal{D}_{skilledForg}\}$. Each of these sets contain M pairs of signature samples, and their label: $\{(X, Y)^{(m)}\}, n = 1...M$. For genuine signatures, the label represents from which user the signature comes from, and for forgeries it represents for which user the forgery was created.

Next, we consider a learning set $\mathcal{L}$, containing samples from a set of users $\mathcal{Y}$. These are the users enrolled to the system, and the learning set consists of the samples obtained in the enrolment process. Therefore, this set only contains the genuine signatures for each user: $\mathcal{L} = \{\mathcal{L}_{genuine}\}$.

For evaluating the performance of the system, we consider a testing set $\mathcal{T}$, containing samples from the same set of users $\mathcal{Y}$. This set represents the new signatures (genuine and forgeries) presented to the system after it has been trained. This dataset consists of genuine signatures and skilled forgeries: $\mathcal{T} = \{\mathcal{T}_{genuine} \bigcup \mathcal{T}_{skilledForg}\}$.

This problem is then addressed as a Pattern Recognition problem, where the datasets $\mathcal{D}$ and $\mathcal{L}$ are used to train a classifier (estimate the parameters of a model), and then are used to generalize to unseen examples. The quality of the model is evaluated using the testing set $\mathcal{T}$. Commonly, the following steps are taken: (image) preprocessing, feature extraction, followed by training a classifier. These steps are reviewed in detail in the subsequent sections of this report.

Figure 1: Superimposed examples of multiple signatures of the same user. We can notice a high intra-class variability of the signatures of the user [10].

2.1 Challenges

Before presenting the methods proposed to address this task, we first highlight some of the most important challenges of Offline Signature Verification.

One of the main challenges for the task is having a high intra-class variability. Compared to physical biometric traits, such as fingerprint or iris, handwritten signatures from the same user often show a large variability between samples. This problem is illustrated in Figure 1. This issue is aggravated with the present of low inter-class variability, when we consider differences between genuine signatures and skilled forgeries. Figure 2 shows some examples of genuine signatures and forgeries from the GPDS dataset [9]. In these examples, we can see that not only genuine signatures from the same user can be very different one from another, but that in some cases the skilled forgeries have a great degree of resemblance to some genuine samples.

Another important challenge for training an automated signature verification system is the presence of partial knowledge during training. In a realistic scenario, during training we only have access to genuine signatures for the users enrolled to the system. During operations, however, we not only want the system to be able to accept genuine signatures, but to reject forgeries. This is a challenging task, since during training a classifier has no information to learn what exactly distinguishes a genuine signature and a forgery for the users enrolled in the system.

Lastly, the amount of data available for training is often limited in real applications. During the enrolment phase, users are often required to supply only a few samples of their signatures.

3 Datasets

A large amount of research in automated signature verification has been conducted with private datasets. This makes it difficult to compare relate work, since an improvement in classification performance may be attributed to a better method, or simply to a cleaner or simpler database. In the last decade, however, a few signature datasets were made available publicly for the research community, addressing this gap.

The process to acquire the signature images follows similar steps for most of the datasets. The genuine signatures are collected in one or more sessions, and require the user to provide several samples of their signatures. The user receives a form containing many cells, and provide a sample of his/her signature in each cell. The cells often have sizes to match common scenarios such as bank cheques and credit card vouchers [9]. The collection of forgeries follows a different process: the users receive samples from genuine signatures and are asked to imitate the signature one or more times. For some datasets, the same set of users that provided genuine signatures are used to provide forgeries (of other users' signatures), while in other datasets the forgeries were created using a different set of users. It is worth noting that the users that provide the forgeries are not experts in producing forgeries. Figure 3 illustrate the forms used to collect genuine signatures and forgeries for one of these datasets. After the forms are collected, they are scanned (often at 300 dpi or 600 dpi), and pre-processed. The preprocessing consists in separating the signatures into individual images, and for some datasets other actions are taken as well, such as binarization or noise removal.

Table 1 presents a summary of the most commonly used signature datasets. The CEDAR dataset [11] was created with data from 55 users, that wrote their signatures in predefined spaces of 2 x

Figure 2: Examples of genuine signatures and skilled forgeries from the GPDS-160 dataset. Left: three genuine samples for each user. Right: a skilled forgery for the same user [8].

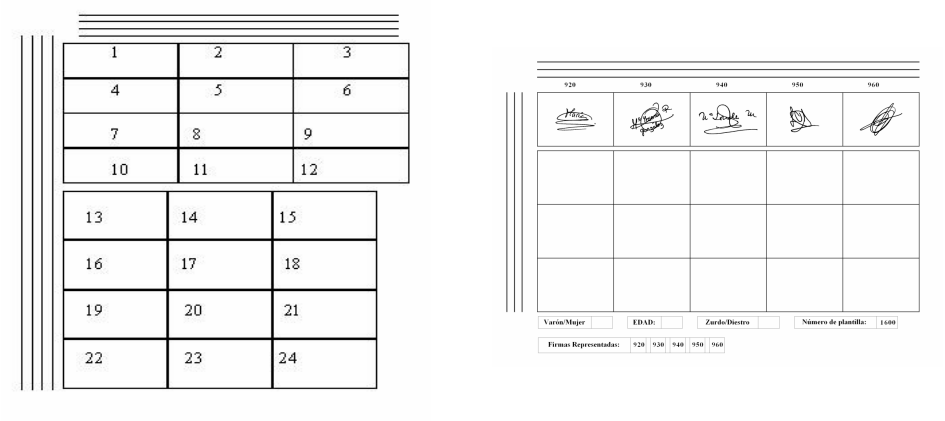

Figure 3: Sample forms used for collecting samples for the GPDS-960 signature dataset. Left: Form used to collect genuine samples, in big and small boxes. Right: Form used to collect forgeries, containing 5 random genuine signature for the person to imitate 3 times [9].

Table 1: Commonly used signature datasets

Dataset Name	Users	Genuine signatures	Forgeries
Brazilian (PUC-PR)	60 + 108	40	10 simple, 10 skilled[2]
CEDAR [11]	55	24	24
MCYT-75 [13]	75	15	15
MCYT-100 [12]	100	25	25
GPDS Signature 160 [14]	160	24	30
GPDS Signature 960 [9]	960	24	30
GPDS Signature 960 Grayscale [9]	881	24	30

2 inches. At random, users were asked to create forgeries for signatures from other writers. Data was collected during three sessions to provide more variability for the signatures. For the MCYT datasets [12], [13], the signatures were captured in an online (dynamic) and offline (static) formats. Users provided their signatures using a pen and paper templates over a pen tablet, and both the online information (e.g. position over time) as the offline information (signature image) were captured. For the GPDS datasets [9], [14], the genuine signatures were taken in a single session, using the form presented in figure 3. A total of 1040 users provided their signatures. The analysis of the forms showed several cases where the users' signatures crossed the borders of the forms, resulting in the forms from 80 people being discarded (leaving data from 960 users). A different set of users (1920 people) participated in the creation of the forgeries. Forgers were allowed as much time as they wanted to produce the them. It is worth noting that some of the signatures of this dataset are no longer available as they were lost during a move - leaving signatures from 881 users available to use (details in [1]).

4 Preprocessing

As with most pattern recognition problems, preprocessing plays an important role in signature verification. Signature images may present variations in terms of pen thickness, scale, rotation, etc., even among authentic signatures of a person.

Common preprocessing techniques are: signature extraction, noise removal, application of morphological operators, size normalization, centering and binarization [3].

- **Signature extraction** - This is an initial step that consists in finding and extracting a signature from a document. This is a particular challenging problem in bank cheques, where the signature is often written on top of a complex background [15], [16]. This step is, however, not considered in most signature verification studies, that already consider signatures extracted from the documents, that is, each input image containing a single signature, commonly with no background and little noise.
- **Noise Removal** - The digital version of the signature images is commonly obtained by the use of scanners, and this process may produce noise in the signatures, such as single black pixels on a white background or single white pixels on a black background. A common strategy is to apply a noise removal filter to the image, such a median filter [17]. It is also common to apply morphological operations to fill small holes and remove small regions of connected components (e.g. less than 20 black pixels on the component, in a binarized image) [17] [8].
- **Size normalization and centering** - Depending on the properties of the features to be used, a few strategies for size normalization have been adopted. The simplest strategy is just to crop the signature images to have a "tight fit" on the signature, that is, crop the image, such that the frame touches the signature in the four directions (left, right, top, bottom) [18].

[1] http://www.gpds.ulpgc.es/download/

[2] for 60 users only

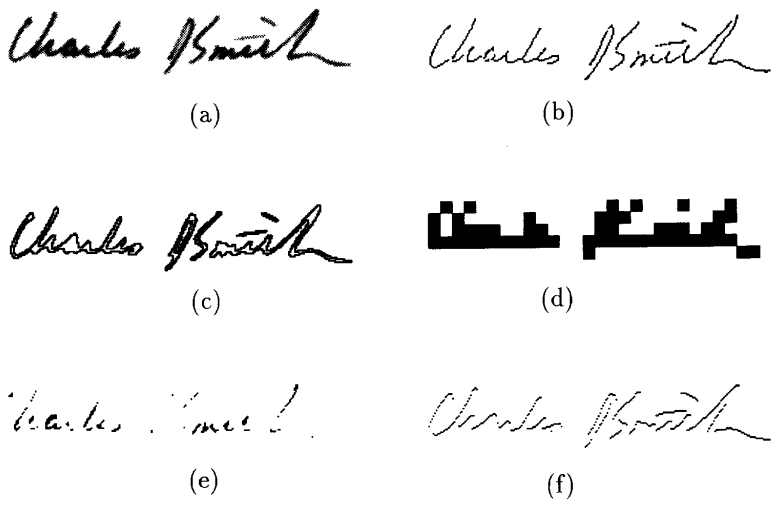

Figure 4: Signature representations. (a) original, (b) skeleton, (c) outline, (d) ink distribution, (e) high pressure regions, (f) directional frontier. Adapted from [17]

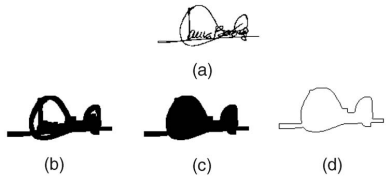

Figure 5: Signature representations. (a) original, (b) dilated, (c) filled, (d) outline of the signature [14]

Another strategy is to user an even narrower bounding box, such as cutting strokes that are far from the image centroid, that are often subject to more variance in a user's signature [8]. Other authors set a fixed desired width, keeping the height-to-width ratio unchanged [19], or setting a fixed frame size (width and height), and centering the signature in this frame [20]. When setting a fixed width and height, the signature is usually centered: the centroid of the signature is calculated and used to center the image, adding white borders to fill the remainder of the image [17].

- **Signature representation** - Besides just using the gray-level image as input, other representations for the signature have been considered. Huang et al. [17] consider the signature's skeleton, outline, ink distribution, high pressure regions and directional frontiers. An example of these different representations is found in Figure 4. Ferrer et al [14] used a morphological operator to dilate the signature image and fill it, to obtain an outline (Figure 5)

- **Signature Alignment** - alignment is a common strategy in online signature verification, but not broadly applied for the offline scenario. Yilmaz [8] propose aligning the signatures for training, by applying rotation, scaling and translation. The objective is to maximize the similarity of each pair of signatures, by minimizing the distance between their feature vectors, after applying the transformations. The best parameters (rotation angle, scaling factor and number of pixels in the translation) are found by an exhaustive search. Yilmaz reported increased accuracy by aligning the signatures during training, but found that aligning the queries during testing did not improve accuracy, while increasing the computational cost. Kalera et al. [11] propose a method to perform Rotation normalization, using first and second order moments of the signature image.

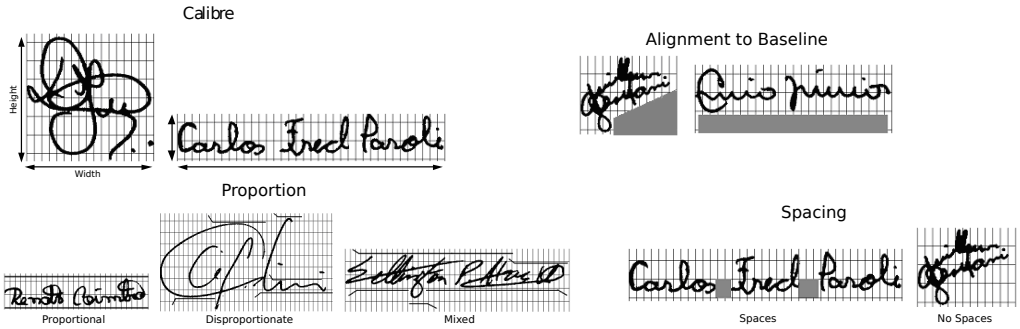

Figure 6: Examples of static graphometric features [22]

5 Feature Extraction

After the signatures have been acquired and pre-processed, the next step is to extract discriminant features from the signature images. In the last decades, offline signature verification has been studied from many perspectives, yielding multiple alternatives for feature extraction. Broadly speaking, the feature extraction techniques can be classified as **Static** or **Pseudo-dynamic**. Using static features is the most straightforward approach, given the nature of the offline signature verification problem. Pseudo-dynamic features attempt to recover dynamic information from the signature execution process (such as speed, pressure, etc.). Another broad categorization of the feature extraction methods is between **Global** and **Local** features. Global features describe the signature images as a whole - for example, features such as height, width of the signature, or in general feature extractors that are applied to the entire signature image. In contrast, local features describe parts of the images, either by segmenting the image (e.g. according to connected components) or most commonly by the dividing the image in a grid (of Cartesian or polar coordinates), and applying feature extractors in each part of the image.

The following sections describe the most common feature descriptors used in the literature.

5.1 Geometric Features

Geometric features measure the overall shape of a signature. This category includes basic descriptors, such as the signature height, width, caliber (height-to-width ration) and area. More complex descriptors include the count of endpoints and closed loops [19]

Besides using global descriptors, several authors also generate local geometric features by dividing the signature in a Cartesian grid and calculating features from each cell of the grid. For example, using the pixel density within grids ([19], [21], [10]). In particular, Huang and Yan [17] used several types of signature representations (outline, skeleton, etc.) as input, divided each representation in a grid format, and used the pixel density in each cell of the grids as the features for the signature.

5.2 Graphometric features

Forensic document examiners use the concepts of graphology and graphometry to examine handwriting for several purposes, including detecting authenticity and forgery. Oliveira et al. [22] investigated applying such features for automated signature verification. From the features used in graphometry studies, they selected a subset that could be applied to the task (e.g. features that could be described algorithmically), and proposed a set of feature descriptors. They considered the following static features: **Calibre** - the ratio of Height / Width of the image; **Proportion**, referring to the symmetry of the signature, **Alignment to baseline** - describing the angular displacement to an horizontal baseline, and **Spacing** - describing empty spaces between strokes. Examples of these features can be found in Figure 6.

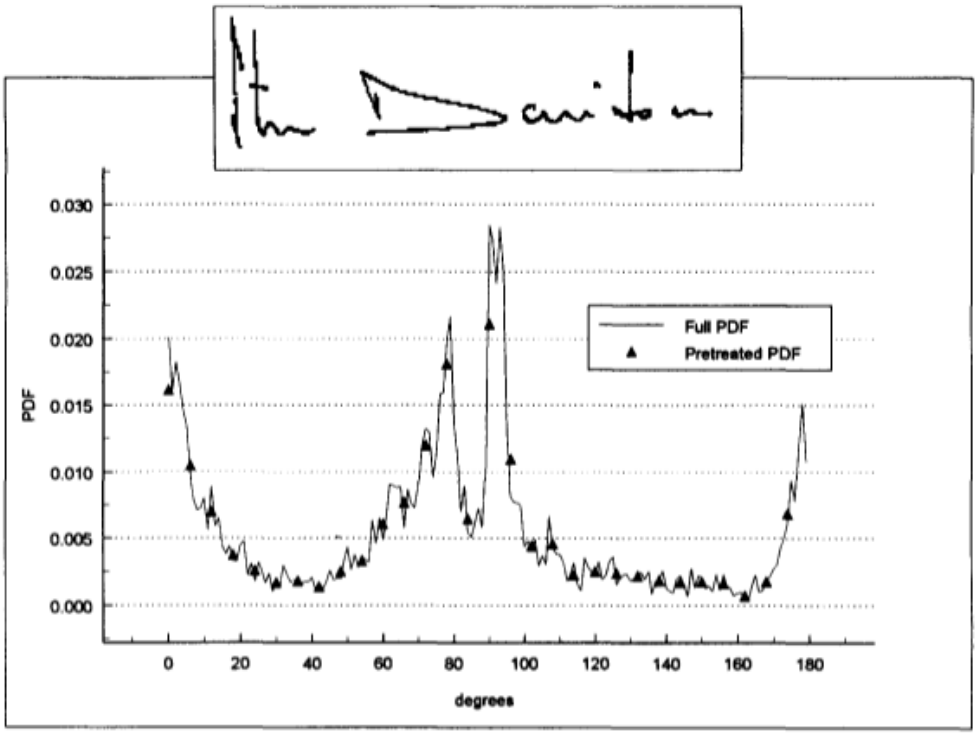

Figure 7: Example of a directional PDF extracted from a signature image. We can notice peaks close to 0, 180 and 90 degrees - showing the predominance of horizontal and vertical strokes for this signature. [24]

5.3 Directional features

Directional features seek to describe the image in terms of the direction of the strokes in the signature. Sabourin [23] and Drouhard [24] extracted Directional-PDF (Probability Density Function) from the gradient of the signature outline (see Figure 7). Recent work from Rivard [25] used this method of feature extraction using grids of multiple scales, yielding promising results.

Zhang et al. have investigate the usage of pyramid histogram of oriented gradients (PHOG) [26]. This descriptor represents local shapes in a image by a histogram of edge orientations, also in multiple scales (Figure 8). This strategy obtained state-of-the-art results on an experiment that used skilled forgeries for training.

5.4 Mathematical transformations

Researchers have used a variety of mathematical transformations as feature extractors. Nemcek and Lin [27] investigated the usage of a fast **Hadamart** transform and spectrum analysis for feature extraction. Pourshahabi et al. [20] used a **Contourlet** transform as feature extraction, stating that it is an appropriate tool for capturing smooth contours. Coetzer et al. [28] used the discrete **Radon** transform to extract sequences of observations, for a subsequent HMM training. Deng et al [29] proposed a signature verification system based on the **Wavelet** transform: first the image is preprocessed to obtain a closed-contour of the signature. For each pixel in the contour, the coordinates (x and y), and the tangential angle are recorded. Each sequence is represented as a one-dimensional signal, and then a discrete wavelet transform is used to decompose the signal. Zouari et al [30] has investigate the usage of the **Fractal** transform for the problem.

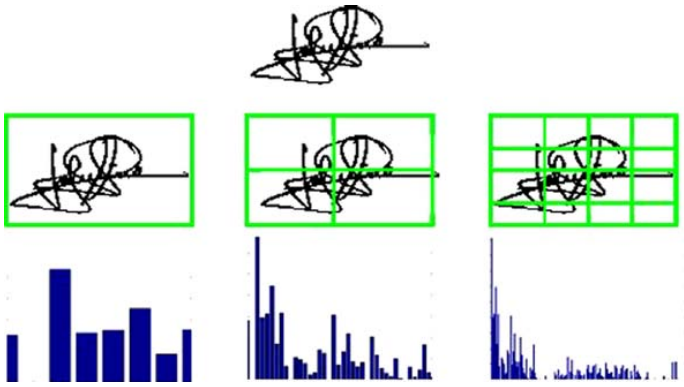

Figure 8: PHOG features - the descriptor consists of histogram of oriented gradients extracted at different resolutions [26]

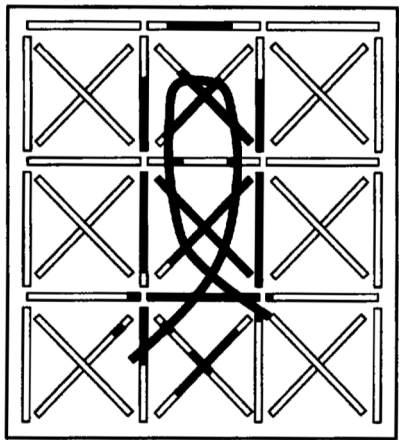

Figure 9: Example of the application of Extended Shadow Code [23]

5.5 Shadow-code

Sabourin et al. [23], [31] proposed an Extended Shadow Code for signature verification. A grid is overlaid on top of the signature image, containing horizontal, vertical and diagonal bars, each bar containing a fixed number of bins. Each pixel of the signature image is then projected to its closest bar in each direction (i.e. its "shadow" in the vertical / horizontal / diagonal bars), activating the respective bin. An example of this operator is presented in Figure 9. After all pixels are projected to the respective bars, the number of active bins on each bar is counted, and used as a description of the signature. This feature extractor has been used by Rivard [6] and Eskander [7] with multiple resolutions, together with directional features, to achieve promising results on writer-independent and writer-dependent classification, respectively.

5.6 Texture features

Texture features, in particular variants of Local Binary Patterns (LBP), have been used in many experiments in recent years. The LBP operator was introduced by Ojala [32] as a discriminant feature extractor for texture images. The original LBP operator is illustrated in Figure 10 - the objective is to determine the pattern of the local neighborhood of each pixel. Each of the 8 neighboring positions is associated a code that goes from 2^0 to 2^7. The LBP code for the neighborhood of a pixel is then found by adding the codes of all neighbors that have a pixel value larger than the central pixel. Commonly, the LBP codes for all pixels in the image are calculated, and histograms of these codes are

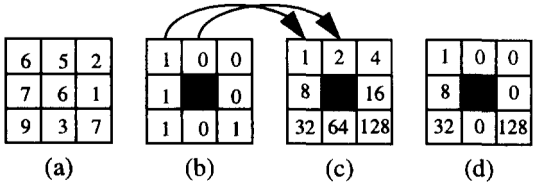

Figure 10: Example application of the original LBP operator [32]. In this example, the LBP code for the central pixel is $1 + 8 + 32 + 128 = 169$

used as features. Several extensions of LBP have been proposed, most commonly a derivation that considers circular neighbors, equally spaced [33]. In this case, the LBP code has two parameters: P and R, where P is the number of neighbors, and R is the radius (distance from the central point). On top of this derived version of LBP, several extensions have been proposed, for instance to group multiple LBP codes together, as to achieve rotation invariance.

In the context of signature verification, LBP has been used by Yilmaz ([34], [8]), together with other descriptors to achieve state-of-the-art results on the GPDS dataset, considering models that do not use skilled forgeries for training. Serdouk et. al [35], [36] used an Orthogonal Combinational LBP and Rotation Invariant LBP features to obtain state-of-the-art results in the GPDS dataset, using skilled forgeries for training the models.

5.7 Interest point matching

Interest point matching methods, such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features) has been largely used for computer vision tasks, such as object recognition and 3D reconstruction. Some authors have created feature extraction techniques on top of these methods, to enable their use in the Signature verification task.

Ruiz-del-Solar et al. [37] used SIFT to extract local interest points from both query and reference samples to build a writer-dependent classifier. After extracting interest points from both images, they generated a set of 12 features, using information such as the number of SIFT matches between the two images, the processing time (as a measure of complexity of the matching process), along with other information from the transformations. They then trained a Bayes classifier with these features. Using skilled forgeries for training, they obtained good results on the GPDS dataset.

Malik et al. [38] used SURF features to classify among genuine signatures, forgeries and disguised signatures. They first used SURF to extract interest points in the signature images, and used these features to assess the local stability of the signatures (i.e. find parts of the genuine signatures that are more stable over time). During classification, only the stable interest points are used for matching. The number of keypoints in the query image, and the number of matched keypoints were used to classify the signature as genuine or forgery.

5.8 Pseudo-dynamic features

Oliveira et al. [22] presented a set of pseudo-dynamic features, based on graphometric studies: **Distribution of pixels**, **Progression** - that measures the tension in the strokes, providing information about the speed, continuity and uniformity, **Slant** and **Form** - measuring the concavities in the signature. These feature extractors are illustrated in Figure 11.

More recently, Bertolini et al. [39] proposed a new descriptor that considers the **curvature** of the signature. This was accomplished by fitting Benzier curves to the signature outline (more specially, to the largest segment of the signature), and using the parameters of the curves as features.

5.9 Feature learning

In recent years, there has been an increased interest on techniques that do not rely on hand-engineered feature extractors. Instead, the idea is to learn feature representations from *raw* data

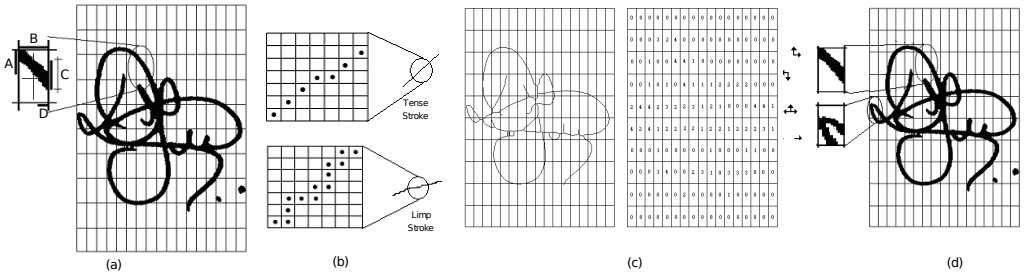

Figure 11: Pseudo-dynamic Graphometric features. (a) Distribution of pixels, (b) Progression, (c) Slant, (d) Form [22].

(pixels, in the case of images). This is the case of Deep Learning models, as reviewed by Bengio in [40] and [41].

Although these techniques have been widely used in recent years for many computer vision tasks, they have not been broadly used for signature verification. Murshed et al. [42], [43], used auto-encoders (called Identity-Mapping Backpropagation in their work) to perform image compression (dimensionality reduction) followed by a Fuzzy ARTMAP classifier. This work, however, considers only a single hidden layer, with less units than the input. In contrast, in recent successful applications of auto-encoders, multiple layers of representations are learned, often in an over-complete format (more hidden units than visible units), where the idea is not to reduce dimensionality, but "disentangle" the factors of variation in the inputs [40]. Ribeiro et al [44] used RBMs to learn a representation for signatures, but only reported a visual representation of the learned weights, and not the results of using such features to discriminate between genuine signatures and forgeries. Khalajzadeh [45] used CNNs for Persian signature verification, but only considered random forgeries in their tests.

6 Model Training

We now discuss the actual machine learning models used for the task of signature verification. The models can be broadly classified in two groups: **writer-dependent** and **writer-independent**. In the first case, that is more common in the literature, a model is trained for each user, using the user's genuine signatures, and commonly random forgeries (by using genuine signature from other users). During the testing phase, a new signature is input to the model, that classifies a sample as genuine or forgery. The writer-independent approach, on the other hand, involves only a single classifier for all users. In this case, usually a distinct set of users is used for training and testing, and during the test phase both the model is used, as well as reference genuine samples for each user, to make a decision (genuine or forgery).

Some authors use a combination of both approaches. For example, Eskander et al [7] trained a hybrid writer-independent-writer-dependent solution, where a writer-independent classifier is used for classification until a reasonable number of genuine samples for the user is obtained - at this point a writer-dependent classifier is trained and used for subsequent queries. Yilmaz [8] propose a hybrid approach, where the results of both a writer-independent and writer-dependent classifiers are combined. As a third scenario, some authors train writer-dependent classifiers, but due to the low number of samples, optimize the models' hyperparameters in a writer-independent format (as in [46]).

Besides the most basic classifiers (e.g. simple thresholding and nearest-neighbors), several strategies have been tried for the task of signature verification. The following sections cover the main models used for the task.

6.1 Neural Networks

Neural Networks have been explored by some authors to perform signature verification, in particular for writer-dependent classification. In this case, a dataset for each user is created using genuine samples, and some type of forgery (usually random forgeries, by using genuine samples from other

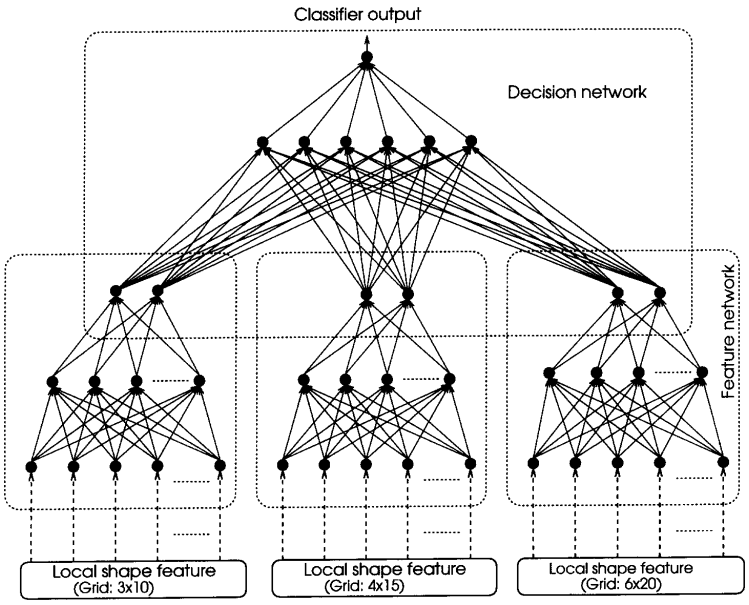

Figure 12: The neural network used by Huang and Yan [17]

users). Features are extracted using any feature extraction mechanism, and this dataset is then used to train a neural network. Huang and Yan [17] used Neural Networks to classify between genuine signatures and random and targeted forgeries. They trained multiple networks on features extracted at different resolutions (grid sizes), and another network to make a decision, based on the outputs of these networks (see Figure 12). Shekar et al [47] trained neural networks and support vector machines obtaining state of the art results on the GPDS dataset.

Murshed et al. [43] used a Fuzzy ARTMAP architecture for the problem. For each user, the signatures were divided in a cartesian grid, and only the cells (patches) that contained signature pixels were considered (i.e. forming a particular "mask" for each user). For each cell, they first compressed the image patch using an auto-encoder. The compressed patch was then fed to a Fuzzy ARTMAP architecture, that learned the variations in the writer's signatures. The verification process consisted of two stages, starting with a global approach, that considered how much of the query signature lied outside of the grid, and how much of the grid was covered (given the "mask" for the target user). If the query signature was not consistent with this grid, it was rejected. Otherwise, the system employed a second step, that averaged the predictions of the Fuzzy ARTMAPs, to obtain a final prediction.

6.2 Hidden Markov Models

Several authors have proposed using Hidden Markov Models for the task of signature verification [10], [22], [46]. HMMs are generative models that attempt to learn the joint distribution $P(X, Y)$, where X are the features and Y are the labels, instead of just the conditional distribution $P(Y|X)$. In particular, HMMs with a left-to-right topology have been mostly studied, as they match the dynamic characteristics of American and European handwriting (with hand movements from left to right).

In the work from Justino [10], Oliveira [22] and Batista [46], the signatures are divided in a grid format. Each column of the grid is used as an observation of the HMM, and features are extracted from the different cells within each column - that is, a signature image I is converted to a sequence of feature vectors $\mathcal{F} = \{f^1, ..., f^C\}$, where C is the number of columns in the grid (observations). To quantize (discretize) the sequence of observations, commonly the K-means algorithm is used on the feature vectors (of a subset of the dataset) to form a a *codebook* $\mathcal{Q} = \{Q^1, ..., Q^k\}$. The sequence of the feature vectors for a signature image is then quantized using this codebook, forming a sequence of observations $\mathcal{O} = \{O^1, ..., O^C\}$, where each observation is a symbol from the codebook $O^i \in \mathcal{Q}$.

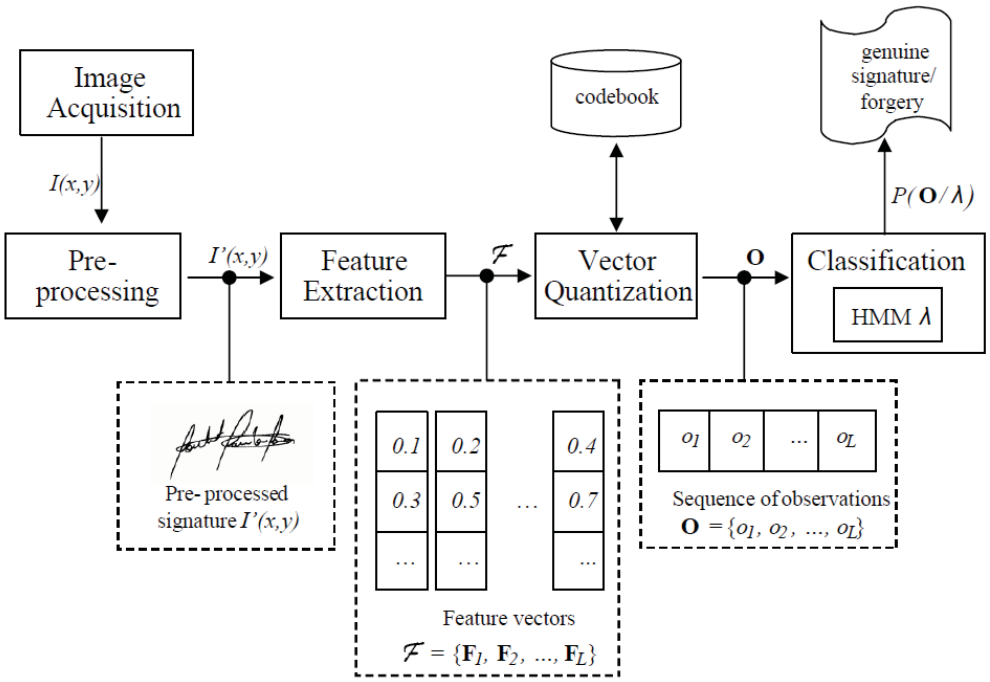

Figure 13: The steps for training an HMM for signature verification [49]

With the sequence of discrete observations, a HMM λ is trained, commonly using the Baum-Welch Forward-Backward algorithm [48]. This process is illustrated in Figure 13. In the verification phase, a sequence of feature vectors is extracted from the signature and quantized using the codebook. The HMM is then used to calculate the likelihood of the observations given the model $P(\mathcal{O}|\lambda)$. After calculating the likelihood, a simple threshold can be used to discriminate between genuine signatures and forgeries [10], or the likelihood itself can be used for more complex classification mechanisms, such as the work from Batista [46] that used HMMs trained on the genuine class, and HMMs trained on the forgery (random forgery) class, and used the likelihoods obtained by the different HMMs as a feature vector for another classifier to make the decision.

6.3 Support Vector Machines

In its original formulation, SVMs are used for two-class classification problems, and learn a hyperplane that maximizes the margin, that is, the distance between the hyperplane and the samples of each class closest to this hyperplane. A second formulation of the model considers a "soft margin", to enable the classification of problems that are not linearly separable [50]. The effectiveness of the model is further increased by the usage of the "kernel trick", that allows the classification in a higher-dimensional feature space implicitly, by the usage of a kernel function.

Support Vector Machines have been extensively used for signature verification, for both writer-dependent and writer-independent classification [51], [52], [39], [53]. In recent years, Guerbai et al [54] used One-Class SVMs for the task. This type of model attempt to only model one class (in the case of signature verification, only the genuine signatures), which is a desirable property, since for the actual users enrolled in the system we only have the genuine signatures to train the model. However, even in this scenario there is a need to use forgeries (in their case, random forgeries) to define the thresholds of the classifiers.

6.4 Writer-independent Classification

Some authors have adopted a writer-independent approach for the signature verification problem [6], [7], [39], [53]. The objective is to train a classifier on a set of the users, and then use this

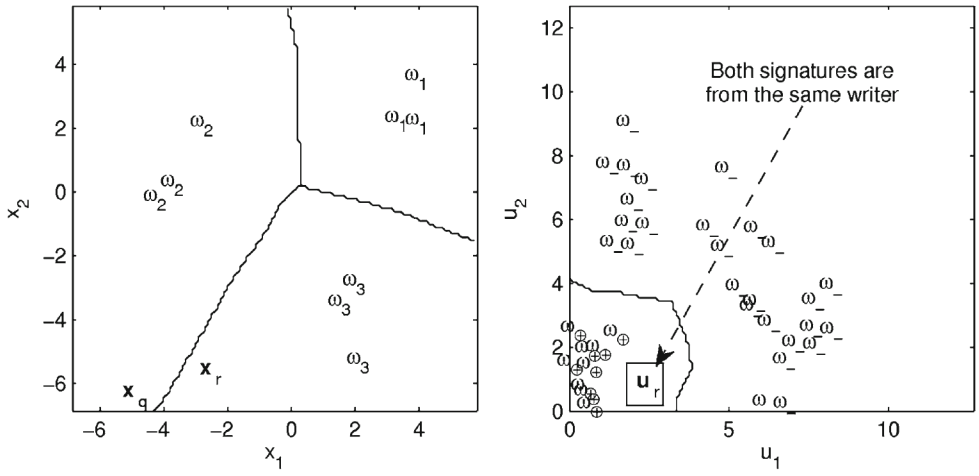

Figure 14: Example of the dissimilarity representation. Left: samples from three classes in the feature space (with two dimensions). Right: samples in the dissimilarity representation, where positive samples ω_+ come from the same class (e.g. ω_1 and ω_1), while negative samples ω_- are from different classes (e.g. ω_1 and ω_2). For a new class c, given a reference point x_r and a query point x_q, a model trained in the dissimilarity representation for users 1-3 can be used to classify samples from the new user. In this case, u_r is calculated as the dissimilarity from x_q and x_r. With the example decision boundary, the point u_r is classified as positive, meaning that the query sample x_q is from the same class c of x_r [25]

classifier in a disjoint set of users. In particular, the usage of the dissimilarity representation (or dichotomy transformation) has shown to be promising for this problem. The idea of the dissimilarity representation is to transform an N-class problem into a 2-class problem. This transformation is applied to pairs of samples in the feature space, obtaining new samples from a positive class (when the two original samples belong to the same class) and negative class (when the two original samples belong to different classes). A common approach for this transformation is to simply use the absolute value of the difference between two feature vectors:

$$u_r = |x_q - x_r| \tag{1}$$

In this case, the resulting feature vector (in the dissimilarity space) has the same cardinality (number of features) as the original samples. This procedure is illustrated in figure 14. One important advantage of this model is that even with a small number of samples in the feature representation, it is possible to generate a large number of samples in the dissimilarity space. This approach also has the benefit of only requiring one model to be trained (instead of one model per user), which is an interesting property for deploying this model in a real application.

6.5 Ensemble of classifiers

Instead of simply training one classifier for the task, some authors have adopted strategies to train multiple classifiers, and combine their predictions when classifying a new sample.

Bertollini et al. [39] used a static ensemble selection with graphometric features, to reduce forgeries in a writer-independent classification. They apply a strategy of "overproduce and choose", by generating a large pool of classifiers (trained with different grid sizes), and used a genetic algorithm to select a subset of the models, building an ensemble of classifiers. They performed this optimization (ensemble selection) using two fitness functions: maximize the Area Under the Curve (AUC) and maximize the True Positive Rate, given a fixed False Positive Rate. This strategy was used to obtain state-of-the-art results on the GPDS and Brazilian datasets.

Batista et al [46] used dynamic selection of classifiers for building a writer-dependent system. First, a bank of HMMs $(M) = \lambda^1, ..., \lambda^N$ is trained, using different number of observations (i.e. dividing the signature image into different number of columns), different codebook sizes, and considering

two types of HMMs: one to model the genuine class (trained with genuine samples), and one to model the forgery class (trained with random forgeries). For a given sample, the posterior likelihood $P(O|\lambda^i)$ is calculated for all HMMs. The set of likelihoods is considered as a feature vector, and a specialized random subspace method (based on [55]) is used to train an ensemble of classifiers (each classifier in a subspace of the features). Two strategies for dynamic selection of classifiers, based on Output Profiles [56], are used, to select which classifiers should be used for a given query signature. After the ensemble is selected, the output of the classifiers in the ensemble is combined using a majority vote, yielding a final classification label for the sample.

6.6 Feature selection

Rivard et al. [25] and Eskander et al. [7] have used a feature selection approach for signature verification. Rivard et al. trained a writer-independent classifier, by first extracting a large number of features from each signature (over 30 thousand features), applying feature extractors at different scales of the image. A Boosting Feature Selection (BFS) approach was then used to simultaneously select features and train a classifier. Their method consisted in training an ensemble of decision stumps (equivalent to a decision tree with only one node), where each decision stump only used one feature. With this approach, they were able to obtain a smaller feature representation (less than a thousand features) that achieved good results in the Brazilian and GPDS datasets. Eskander et al. [7] extended Rivard's work to train a hybrid writer-independent-writer-dependent classifier, by first training this writer-independent classifier to perform feature selection, and then train writer-dependent classifiers using only the features that were selected by the first model. This strategy presented good results when a certain number of samples per user is reached.

6.7 Data augmentation

One of the main challenges for building an automated signature verification system is the low number of samples per user for training. To address this issue, some author have suggested ways to generate more samples, based on existing genuine signatures.

Huang and Yan [17] have proposed a set of "perturbations" to be applied to each genuine signature, to generate new samples: slant, rotation, scaling and perspective. In their work, they considered a set of "slight distortions", used to create new genuine samples, and "heavy distortions" to generate forgeries from the genuine samples. More recently, Ferrer et al [57], [58] have proposed a signature synthesis approach inspired on a neuromotor model.

7 Conclusion

Over the last decade, several researchers have proposed different methods for Offline Signature Verification. In spite of the these advancements, the experimental results still report somewhat large error rates for distinguishing genuine signatures and skilled forgeries, when large public datasets are used for testing, such as GPDS. Error rates are at least around 7-8% in the best reported results, even when the number of samples for training is around 10-15 (results are worse with 1-3 samples per user, which is a common scenario in banks and other institutions). Although these recent results are encouraging, and improve upon previous work, the error rates are still large considering the critical environments where signature verification is used in practice.

Analyzing the recent contributions to the field, we can notice that they concentrate in the following categories:

- **Obtaining better features** - In recent years, several new feature extractors have been proposed for the task. Texture features (LBP variations), interest-point matching (SIFT, SURF) and directional features (HOG) have been successfully used to increase the accuracy of Offline Signature Verification Systems.
- **Improving classification with limited number of samples** - Given the severe constraints in practical applications, researchers have searched for ways to increase performance in cases where a small number of samples per user is available. In particular, the creation of dissimilarity-based writer-independent solutions have shown to be promising to address this problem.

- **Augmenting the datasets** - Related to the problem of having low number of samples per user, some researchers have focused in generating synthetic signature samples, in order to increase the number of samples available for training.
- **Building model ensembles** - In order to increase classification accuracy, and the robustness of the solutions, some researchers have investigated the creation of both static and dynamic ensembles of classifiers.

In the authors' opinion, this trend will continue for future work, with researchers continuing to explore better feature representations for the problem; and investigating solutions that address the particularities of the problem domain, such as having small number of samples per training.

References

[1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 4–20, 2004.

[2] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 63–84, 2000.

[3] D. Impedovo and G. Pirlo, "Automatic signature verification: the state of the art," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 5, pp. 609–635, 2008.

[4] R. Plamondon and G. Lorette, "Automatic signature verification and writer identificationthe state of the art," *Pattern recognition*, vol. 22, no. 2, pp. 107–131, 1989.

[5] F. Leclerc and R. Plamondon, "Automatic signature verification: The state of the art19891993," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 03, pp. 643–660, 1994.

[6] D. Rivard, "Multi-feature approach for writer-independent offline signature verification," Ph.D. dissertation, École de technologie supérieure, 2010.

[7] G. Eskander, R. Sabourin, and E. Granger, "Hybrid writer-independent-writer-dependent offline signature verification system," *IET Biometrics*, vol. 2, no. 4, pp. 169–181, Dec. 2013.

[8] M. B. Yilmaz, "Offline Signature Verification With User-Based And Global Classifiers Of Local Features," Ph.D. dissertation, Sabanc University, 2015.

[9] J. Vargas, M. Ferrer, C. Travieso, and J. Alonso, "Off-line Handwritten Signature GPDS-960 Corpus," in *Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007*, vol. 2, Sep. 2007, pp. 764–768.

[10] E. J. Justino, A. El Yacoubi, F. Bortolozzi, and R. Sabourin, "An off-line signature verification system using HMM and graphometric features," in *Fourth IAPR International Workshop on Document Analysis Systems (DAS), Rio de*. Citeseer, 2000, pp. 211–222.

[11] M. K. Kalera, S. Srihari, and A. Xu, "Offline signature verification and identification using distance statistics," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 07, pp. 1339–1360, Nov. 2004.

[12] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, and others, "MCYT baseline corpus: a bimodal biometric database," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, 2003.

[13] J. Firrez-Aguilar, N. Alonso-Hermira, G. Moreno-Marquez, and J. Ortega-Garcia, "An off-line signature verification system based on fusion of local and global information," in *Biometric Authentication*. Springer, 2004, pp. 295–306.

[14] M. Ferrer, J. Alonso, and C. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 993–997, Jun. 2005.

[15] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo, "A multi-expert signature verification system for bankcheck processing," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 05, pp. 827–844, 1997.

[16] S. Djeziri, F. Nouboud, and R. Plamondon, "Extraction of signatures from check background based on a filiformity criterion," *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1425–1438, Oct. 1998.

[17] K. Huang and H. Yan, "Off-line signature verification based on geometric feature extraction and neural network classification," *Pattern Recognition*, vol. 30, no. 1, pp. 9–17, Jan. 1997.

[18] S. Ghandali and M. Moghaddam, "A Method for Off-line Persian Signature Identification and Verification Using DWT and Image Fusion," in *IEEE International Symposium on Signal Processing and Information Technology, 2008. ISSPIT 2008*, Dec. 2008, pp. 315–319.

[19] H. Baltzakis and N. Papamarkos, "A new signature verification technique based on a two-stage neural network classifier," *Engineering applications of Artificial intelligence*, vol. 14, no. 1, pp. 95–103, 2001.

[20] M. R. Pourshahabi, M. H. Sigari, and H. R. Pourreza, "Offline handwritten signature identification and verification using contourlet transform," in *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of.* IEEE, 2009, pp. 670–673.

[21] A. El-Yacoubi, E. J. R. Justino, R. Sabourin, and F. Bortolozzi, "Off-line signature verification using HMMs and cross-validation," in *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol. 2. IEEE, 2000, pp. 859–868.

[22] L. S. Oliveira, E. Justino, C. Freitas, and R. Sabourin, "The graphology applied to signature verification," in *12th Conference of the International Graphonomics Society*, 2005, pp. 286–290.

[23] R. Sabourin and J.-P. Drouhard, "Off-line signature verification using directional PDF and neural networks," in *, 11th IAPR International Conference on Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, Proceedings*, Aug. 1992, pp. 321–325.

[24] J.-P. Drouhard, R. Sabourin, and M. Godbout, "A neural network approach to off-line signature verification using directional PDF," *Pattern Recognition*, vol. 29, no. 3, pp. 415–424, 1996.

[25] D. Rivard, E. Granger, and R. Sabourin, "Multi-feature extraction and selection in writer-independent off-line signature verification," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 1, pp. 83–103, 2013.

[26] B. Zhang, "Off-line signature verification and identification by pyramid histogram of oriented gradients," *International Journal of Intelligent Computing and Cybernetics*, vol. 3, no. 4, pp. 611–630, 2010.

[27] W. F. Nemcek and W. C. Lin, "Experimental Investigation of Automatic Signature Verification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-4, no. 1, pp. 121–126, Jan. 1974.

[28] J. Coetzer, "Off-line signature verification," Ph.D. dissertation, Stellenbosch: University of Stellenbosch, 2005.

[29] P. S. Deng, H.-Y. M. Liao, C. W. Ho, and H.-R. Tyan, "Wavelet-Based Off-Line Handwritten Signature Verification," *Computer Vision and Image Understanding*, vol. 76, no. 3, pp. 173–190, Dec. 1999.

[30] R. Zouari, R. Mokni, and M. Kherallah, "Identification and verification system of offline handwritten signature using fractal approach," in *Image Processing, Applications and Systems Conference (IPAS), 2014 First International*, Nov. 2014, pp. 1–4.

[31] R. Sabourin and G. Genest, "An extended-shadow-code based approach for off-line signature verification. I. Evaluation of the bar mask definition," in *Conference on Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International*, vol. 2, Oct. 1994, pp. 450–453 vol.2.

[32] T. Ojala, M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[33] T. Menp, "The local binary pattern approach to texture analysis extensions and applications," 2003.

[34] M. B. Yilmaz, B. Yanikoglu, C. Tirkaz, and A. Kholmatov, "Offline signature verification using classifier combination of HOG and LBP features," in *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE, 2011, pp. 1–7.

[35] Y. Serdouk, H. Nemmour, and Y. Chibani, "Combination of OC-LBP and Longest Run Features for Off-Line Signature Verification," in *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, Nov. 2014, pp. 84–88.

[36] ——, "Orthogonal Combination and Rotation Invariant of Local Binary Patterns for Off-line Handwritten Signature Verification," 2014.

[37] J. Ruiz-del Solar, C. Devia, P. Loncomilla, and F. Concha, "Offline Signature Verification Using Local Interest Points and Descriptors," in *Progress in Pattern Recognition, Image Analysis and Applications*, ser. Lecture Notes in Computer Science, J. Ruiz-Shulcloper and W. G. Kropatsch, Eds. Springer Berlin Heidelberg, 2008, no. 5197, pp. 22–29.

[38] M. I. Malik, M. Liwicki, A. Dengel, S. Uchida, and V. Frinken, "Automatic Signature Stability Analysis and Verification Using Local Features," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 621–626.

[39] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers," *Pattern Recognition*, vol. 43, no. 1, pp. 387–396, Jan. 2010.

[40] Y. Bengio, "Learning Deep Architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[41] ——, "Deep learning of representations: Looking forward," in *Statistical Language and Speech Processing*. Springer, 2013, pp. 1–37.

[42] N. A. Murshed, F. Bortolozzi, and R. Sabourin, "Binary image compression using identity mapping backpropagation neural network," in *Electronic Imaging'97*. International Society for Optics and Photonics, 1997, pp. 29–35.

[43] N. A. Murshed, R. Sabourin, and F. Bortolozzi, "A cognitive approach to off-line signature verification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 05, pp. 801–825, 1997.

[44] B. Ribeiro, I. Gonalves, S. Santos, and A. Kovacec, "Deep learning networks for off-line handwritten signature recognition," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2011, pp. 523–532.

[45] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Persian Signature Verification using Convolutional Neural Networks," in *International Journal of Engineering Research and Technology*, vol. 1. ESRSA Publications, 2012.

[46] L. Batista, E. Granger, and R. Sabourin, "Dynamic selection of generativediscriminative ensembles for off-line signature verification," *Pattern Recognition*, vol. 45, no. 4, pp. 1326–1340, Apr. 2012.

[47] B. H. Shekar, R. K. Bharathi, J. Kittler, Y. Vizilter, and L. Mestestskiy, "Grid structured morphological pattern spectrum for off-line signature verification," in *2015 International Conference on Biometrics (ICB)*, May 2015, pp. 430–435.

[48] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[49] L. B. Batista, "Multi-classifier systems for off-line signature verification," Ph.D. dissertation, École de technologie supérieure, 2011.

[50] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[51] E. zgndz, T. entrk, and M. E. Karslgil, "Off-line signature verification and recognition by support vector machine," in *European signal processing conference, EUSIPCO*, 2005.

[52] E. J. R. Justino, F. Bortolozzi, and R. Sabourin, "A comparison of SVM and HMM classifiers in the off-line signature verification," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1377–1385, Jul. 2005.

[53] R. Kumar, J. D. Sharma, and B. Chanda, "Writer-independent off-line signature verification using surroundedness feature," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 301–308, Feb. 2012.

[54] Y. Guerbai, Y. Chibani, and B. Hadjadji, "The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters," *Pattern Recognition*, vol. 48, no. 1, pp. 103–113, Jan. 2015.

[55] T. K. Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 832–844, 1998.

[56] P. R. Cavalin, R. Sabourin, and C. Y. Suen, "Dynamic selection of ensembles of classifiers using contextual information," in *Multiple Classifier Systems*. Springer, 2010, pp. 145–154.

[57] M. Ferrer, M. Diaz-Cabrera, and A. Morales, "Synthetic off-line signature image generation," in *2013 International Conference on Biometrics (ICB)*, Jun. 2013, pp. 1–7.

[58] ——, "Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 667–680, Mar. 2015.