OFFLINE SIGNATURE VERIFICATION WITH USER-BASED AND GLOBAL CLASSIFIERS OF LOCAL FEATURES

by

MUSTAFA BERKAY YILMAZ

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Sabancı University
February 2015

OFFLINE SIGNATURE VERIFICATION WITH
USER-BASED AND GLOBAL CLASSIFIERS
OF LOCAL FEATURES

APPROVED BY

Assoc. Prof. Dr. Berrin YANIKOĞLU .....
(Thesis Advisor)

Assoc. Prof. Dr. Hakan ERDOĞAN .....

Assist. Prof. Dr. Kamer KAYA .....

Prof. Dr. Bülent SANKUR .....

Assist. Prof. Dr. Devrim ÜNAY .....

DATE OF APPROVAL: .....

©Mustafa Berkay Yılmaz 2015
All Rights Reserved

to my family

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Berrin Yanıkoğlu for her endless guidance and support. I would like to thank Assoc. Prof. Dr. Hakan Erdoğan, Assist. Prof. Dr. Kamer Kaya, Prof. Dr. Bülent Sankur and Assist. Prof. Dr. Devrim Ünay for their valuable advices and time.

I am sincerely grateful to my family; my mother, my father, my grandparents for always loving, supporting and motivating me.

OFFLINE SIGNATURE VERIFICATION WITH
USER-BASED AND GLOBAL CLASSIFIERS
OF LOCAL FEATURES

MUSTAFA BERKAY YILMAZ

CS, Ph.D. Thesis, 2015

Thesis Advisor: Berrin YANIKOĞLU

Keywords: offline signature, histogram of oriented gradients, local binary patterns, scale invariant feature transform, user-dependent/indepent classifiers, support vector machines, user-based score normalization

Abstract

Signature verification deals with the problem of identifying forged signatures of a user from his/her genuine signatures. The difficulty lies in identifying allowed variations in a user's signatures, in the presence of high intra-class and low inter-class variability (the forgeries may be more similar to a user's genuine signature, compared to his/her other genuine signatures). The problem can be seen as a non-rigid object matching where classes are very similar. In the field of biometrics, signature is considered a behavioral biometric and the problem possesses further difficulties compared to other modalities (e.g. fingerprints) due to the added issue of skilled forgeries.

A novel offline (image-based) signature verification system is proposed in this thesis. In order to capture the signature's stable parts and alleviate the difficulty of global matching, local features (histogram of oriented gradients, local binary patterns) are used, based on gradient information and neighboring information inside local regions. Discriminative power of extracted features is analyzed using support vector machine (SVM) classifiers and their fusion gave better results compared to state-of-the-art. Scale invariant feature transform (SIFT) matching is also used as a

complementary approach. Two different approaches for classifier training are investigated, namely global and user-dependent SVMs. User-dependent SVMs, trained separately for each user, learn to differentiate a user's (genuine) reference signatures from other signatures. On the other hand, a single global SVM trained with difference vectors of query and reference signatures' features of all users in the training set, learns how to weight the importance of different types of dissimilarities. The fusion of all classifiers achieves a 6.97% equal error rate in skilled forgery tests using the public GPDS-160 signature database.

Former versions of the system have won several signature verification competitions such as first place in 4NSigComp2010 and 4NSigComp2012 (the task without disguised signatures); first place in 4NSigComp2011 for Chinese signatures category; first place in SigWiComp2013 for all categories. Obtained results are better than those reported in the literature. One of the major benefits of the proposed method is that user enrollment does not require skilled forgeries of the enrolling user, which is essential for real life applications.

KULLANICI BAZLI VE EVRENSEL YEREL ÖZNİTELİK SINIFLANDIRICILARI İLE ÇEVRİMDIŞI İMZA DOĞRULAMA

MUSTAFA BERKAY YILMAZ

CS, Doktora Tezi, 2015

Tez Danışmanı: Berrin YANIKOĞLU

Anahtar Kelimeler: çevrimdışı imza, yönlü eğimlerin histogramı, yerel ikili örnekleme, ölçekten bağımsız öznitelik dönüşümü, kullanıcı bağımlı/bağımsız sınıflandırıcılar, karar destek makinası, kullanıcı bazlı skor normalizasyonu

Özetçe

İmza doğrulama, bir kişinin gerçek imzalarından yararlanarak taklit imzalarını saptama problemidir. Zorluk, bir kişinin imzalarındaki geçerli çeşitliliği, yüksek sınıf içi ve düşük sınıflararası çeşitliliğin varlığına rağmen tespit etmekte yatar (taklitler, bir kişinin gerçek bir imzasına, aynı kişinin diğer gerçek imzalarından daha fazla benziyor olabilir). Problem, sınıfların birbirlerine çok benzer olduğu bir esnemez-olmayan nesne karşılaştırma gibi görülebilir. Biyometrik alanında imza, davranışsal bir biyometrik olarak kabul edilir ve ek olarak teknik taklit durumundan dolayı probleme parmak izi tanıma gibi diğer yöntemlerden ileri zorluklar hakimdir.

Bu tezde özgün bir çevrimdışı (resim-bazlı) imza doğrulama sistemi önerilmiştir. İmzanın istikrarlı parçalarını yakalamak ve evrensel karşılaştırmanın zorluğunu hafifletmek için, yerel bölgelerdeki eğim ve komşuluk bilgilerini kullanan yerel öznitelikler (yönlü eğimlerin histogramı, yerel ikili örnekleme) kullanılmıştır. Çıkarılan özniteliklerin ayrıştırıcı gücü karar destek makinası (KDM) ile incelenmiş ve kaynaştırma, literatürdekilerden daha iyi sonuç vermiştir. Ölçekten bağımsız öznitelik dönüşüm karşılaştırması da tamamlayıcı bir yaklaşım olarak kullanılmıştır. Sınıflandırıcı eğitimi için, evrensel ve kullanıcı-bazlı olmak üzere iki farklı yaklaşım incelenmiştir. Her kullanıcı için ayrı ayrı eğitilen kullanıcı-bazlı KDMler, bir kişinin referans (ger-

çek) imzalarını diğer imzalardan ayırmayı öğrenir. Diğer taraftan, eğitim kümesindeki tüm kullanıcıların sorgu ve referans imzalarının öznitelikleri arasındaki fark vektörleriyle eğitilen tek bir evrensel KDM, değişik farklılık türlerinin önemlerinin nasıl ağırlıklandırılması gerektiğini öğrenir. Tüm sınıflandırıcıların kaynaştırılması ile halka açık GPDS-160 imza veritabanında, teknik taklitleri sadece testte kullanmak suretiyle %6.97 eşit hata oranı elde edilmiştir.

Sistemin daha önceki sürümleri çeşitli imza doğrulama yarışmalarını kazanmıştır: 4NSigComp2010 ve 4NSigComp2012 yarışmalarında birincilik (kimlik-inkar-etme imzaları olmadan), 4NSigComp2011 yarışmasında Çin imzaları kategorisinde birincilik, SigWiComp2013 yarışmasında tüm kategorilerde birincilik. Elde edilen sonuçlar, literatürde yayınlanan sonuçlardan daha iyi olmuştur. Önerilen yöntemin en büyük avantajlarından birisi, gerçek hayattaki uygulamalara uygun olarak, kullanıcı kaydı sırasında teknik taklit imzalara ihtiyaç duymamasıdır.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Signature Verification

Signature verification aims to verify the identity of a person through his/her chosen signature. Signature is considered to be a behavioral biometric that encodes the ballistic movements of the signer; as such it is difficult to imitate. Compared to physical traits such as fingerprint, iris or face, a signature typically shows higher intra-class and time variability. Furthermore, as with passwords, a user may choose a simple signature that is easy to forge. On the other hand, the signature's widespread acceptance by the public and niche applications (validating paper documents and use in banking applications) make it an interesting biometric.

Depending on the signature acquisition method used, automatic signature verification systems can be classified into two groups: **online** (dynamic) and **offline** (static). A static signature image, generally scanned at a high resolution (e.g. 600 dpi), is the only input to offline systems. Verification of signatures found on bank cheques and vouchers are among important applications for offline systems. An example set of offline signatures is shown in Figure 1.1.

In addition to the signature image, time dimension is also available for dynamically captured signatures that are acquired using pressure sensitive tablets or smart pens. These input devices sample the signature at a high frequency, resulting in a time ordered sequence of signature's trajectory points. An example online signature capturing device is shown in Figure 1.2. Each point is associated with a corresponding acquisition time stamp and a location coordinate, besides other dynamic features such as pressure and pen inclination angles that can be captured subject to

Figure 1.1: An example set of public figures collected from the web.

Figure 1.2: An example online signature capturing device [1].

the hardware used. Online signature verification is generally used for access control and electronic document authentication types of applications. Due to the differences in the input, preprocessing, feature extraction and classification methods used; online and offline systems show significant variations in their approaches, specifically in representation, preprocessing and matching steps.

Offline signature verification can be said to be more challenging compared to online signature verification. While variations among a user's signatures and easy to forge signatures pose a challenge in both cases, dynamic information available in online signatures make the signature more unique and more difficult to forge. In particular, imitating both the shape and dynamic information of an online signature seems to be difficult except for very simple signatures. In contrast, it is possible in some real life situations, for an impostor to trace over a genuine offline signature

and obtain a high quality forgery. Furthermore, the availability of the signature's trajectory also makes it easier for online verification systems to align two signatures and detect differences.

Higher accuracies obtained in online systems also inspired researchers to recover the dynamic information from static images with some success [3]. Applying special techniques, such as conoscopic holography [4], can reveal stroke order and pressure applied by a pen during handwriting. However, these are bulky and very expensive equipments and the process is inefficient in time and difficult to automate. Furthermore, it may fail with certain paper and pen types; thus such an approach is impractical in the context of automatic signature verification.

Signature authentication scenarios are also two-fold: while forensic examiners are interested in verifying the identity of the signer of a document, many companies such as banks are interested in identity control with online or offline signatures, for routine operations. In the latter case called, high throughput and instant response is desired. Such routine operations can be accelerated by an automatic verification system like the one that is proposed in this thesis.

In a biometric authentication system, users are first **enrolled** to the system by registering their biometric samples (in signature verification case, signatures). During verification, a query signature is provided along with a claimed identity; the query is then compared to the reference signatures of the claimed individual. If the calculated dissimilarity is above a certain threshold, the user is rejected, otherwise authenticated.

Two general approaches may be considered for the signature verification problem, though preferred methods vary for online versus offline systems: User-based modeling/discrimination requires one model per user, generally necessitating a large number of references (typically 10+) for which classifiers such as Hidden Markov Models (HMM), or Support Vector Machines (SVM) are often used. In template-based approach, 1 to 5 references of the claimed identity are enough to be used as templates. Distance between the query signature and the template of the claimed identity is investigated. The query is accepted as genuine if the distance is below a threshold or rejected as forgery, otherwise. Many possible features and matching methods are possible based on the task: Dynamic Time Warping (DTW) is success-

fully used in online signature verification [5] where signature trajectory facilitates the registration of signatures. In offline signature verification, local features that are more resilient to variations are more commonly used with various types of classifiers, after rigid or elastic registration of two signatures, as summarized in Section 1.2.

The system performance is generally reported using the **False Rejection Rate (FRR)** of genuine signatures and the **False Acceptance Rate (FAR)** of forgery signatures. Other measures such as the **Equal Error Rate (EER)**, the error rate where both FAR and FRR are equal or the **Distinguishing Error Rate (DER)** which is the average of FAR and FRR are also commonly reported. Reported EER can be expressed as DER, however reported individual FAR and FRR when calculated as DER can not be expressed as EER. Other evaluation measures include FRR at a certain fixed FAR and the **Receiver Operator Characteristics (ROC)** curve which is a graphical plot relating true accept rate (1-FRR) and FAR, obtained at varying acceptance thresholds.

In real life, a forgery may be signed by an imposter who knows about the target user's signature and who may have even studied it with determination to break into the system. On the other extreme, it may also be the case that the imposter does not know the target user's signature or even his/her name. In some intermediate cases, the imposter may only know about the name of the target but not the signature shape. These differences in information about the signature to be forged or the acquired skill level of the forger are important when evaluating a signature verification system: an uninformed or unskilled forgery is much easier to detect compared to a more skilled one.

In parallel with real life scenarios, research databases define two types of forgeries: a **skilled forgery** refers to a forgery which is signed by a person who has had access to some number of genuine signatures and practiced them for some time. Often, the imposter is simply one of the enrolled users who has been asked to forge the signature of another user, since finding real imposters is not feasible.

Similarly a **random forgery** is typically collected from other people's real signatures, simulating the case where the impostor does not even know the name, nor shape of the target signature and hence uses their own in forgery. In this thesis, as in the literature, when the term "forgery" is used without further qualifications,

it may refer to a skilled or random forgery. An **impostor** is then defined as the person who has provided the forgery signature.

Another definition related to signature forgeries is what is called a **disguised signature** which is generated by the user himself with the purpose of denying the ownership of the signature in the future, for instance for withdrawing money from an account and then denying the operation. This category poses a difficult problem that is not yet addressed by researchers; however there is forensic interest in identifying such forgeries as well.

There are some related applications within the domain of signatures. **Signature recognition** refers to the identification of the person by matching given query to the previously stored samples with known identities. No identity is claimed along with the query. **Signature detection** or **spotting** is the problem of automatically detecting the existence and then the exact location of any signature in a document.

1.2 Literature Review

Offline signature verification is a well-researched topic, where many different approaches have been studied. A series of surveys covering advances in the field are available [6–14]. A more up to date overview of proposed works is detailed in a recent work by Coetzer [15]. Here, we review some of the recent research on offline signatures.

Locating the region of interest: The first step before utilizing further applications such as verification or recognition is to extract the signature region of interest from a document. This step is generally skipped in the works that concentrate on biometric applications of signatures thanks to the public offline signature databases. However there are a few studies in the literature that concentrate on signature localization. In most of the cases of real life scenarios, original documents containing the signatures are available. Signature region is extracted and then verification is proceeded.

Relation between handwriting and signature is analyzed by Bouletreau et al. [16]. A method is applied both to handwriting and signature classification that is based on their fractal behavior. The fractal dimension is a measure of the degree of irregularity or of fragmentation of a set, or the measure of the complexity of the

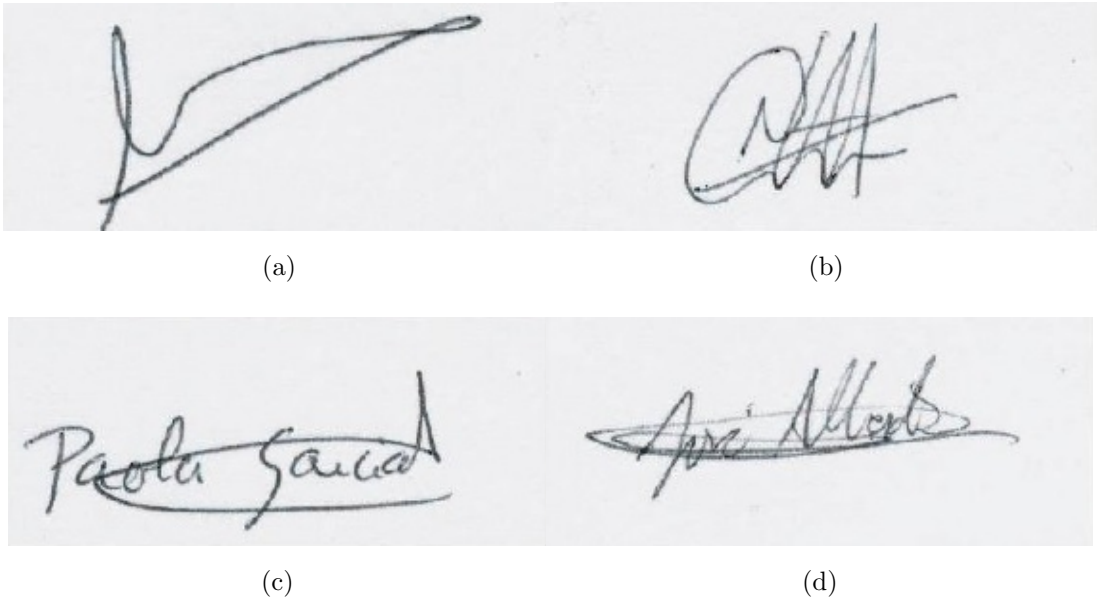
studied set. Different properties related to writing and signature styles are extracted by the help of the method. Properties include cursive writings, legible writings, separated writings. This method provided an evidence of the independence between the behaviors of the writer when he signs and when he writes. Such an independence is reported to have a potential source of enriching information within the context of signature authentication, where the signatures and writings are used as independent identifiers.

Signature region extraction from documents is the main focus of the work by Chalechale et al. [17]. A document image database containing 350 documents signed by 70 different persons who have Persian or Arabic cursive signatures is used. The content of the images include a variety of mixed text of Arabic, Persian and English alphanumeric with different fonts and sizes, a company logo, some horizontal and vertical lines and a cursive signature. The signature region was found correctly in 346 cases (98.86%) and the signature was extracted completely in 342 cases (97.71%). This is due to the fact that some cursive signatures have several disjoint parts while the algorithm focuses on neighboring connected parts.

Recently, a novel method for automated localization of handwritten signatures in scanned documents is proposed by Cüceloğlu and Oğul [18]. The framework is based on the classification of segmented image regions using a set of representative features. The segmentation is done using a two-phase connected component labeling approach. Distinguishing signature and non-signature segments are learnt over a SVM classifier. The experiments on a real banking data set have shown that the framework can achieve a reasonably good accuracy to be used in real life applications.

Determining the signature type: Embellishments, also called flourish, can be defined as the strokes that often begin or end a signature, changing the shape or bounding box significantly. Signatures may be grouped by a signature verification system, based on the complexity of the signature which itself depends on trajectory length and overlap; or the amount of flourish on the signature, in order to handle separate groups differently. Alonso et al. categorize signature according to the amount of embellishments in a signature [2]. Users are categorized according to the type of their signatures as simple flourish (C1), complex flourish (C2), simple flourish

with name (C3), complex flourish with name (C4). Sample signatures from each category are shown in Figure 1.3. Distribution of users in MCYT-75 corpus [19] is found as: C1 (6.67%), C2 (17.33%), C3 (46.67%), C4 (29.33%). With HMM verifier of local information, EERs are sorted from lowest to highest as C4, C2, C3, C1. This is the expected result as complex drawings make the signature harder to imitate and adding the user name information makes it even harder to imitate.



(a) (b)

(c) (d)

Figure 1.3: Sample signatures as categorized by Alonso et al. [2] according to their complexity: simple flourish (a), complex flourish (b), simple flourish with name (c), complex flourish with name (d).

A multi-script signature identification system is offered by Pal et al. [20]. In the proposed signature identification system, the signatures of Bengali (Bangla), Hindi (Devanagari) and English are considered for the identification process. This system identifies whether a claimed signature belongs to the group of Bengali, Hindi or English signatures. SVMs are considered as classifiers for signature identification. A database of 2100 Bangla signatures, 2100 Hindi signatures and 2100 English signatures are used for experimentation. The highest accuracy of 92.14% is obtained based on the gradient features using 4200 (1400 from each language group) samples for training and 2100 (700 from each language group) samples for testing. This approach can be applied with an addition of unknown language class in a real life scenario. By signature language identification, tuned system parameters can be

applied for verification if the queried signature is detected to belong one of the predefined language groups.

Robustness to variations: Genuine signatures contain many variations with respect to illumination, rotation, translation, scaling, pen thickness, embellishments and noise (such as lines or scripts) is an important issue in an image-based biometric. In a work by Nguyen et al., two signatures (query and a reference) are first aligned using rigid or non-rigid alignment and compared based on basic global features extracted from the whole signature (e.g. width/height ratio or pixel density) [21]. This alignment is hoped to compensate for rotation, translation and scaling variations.

Ferrer et al. analyze the robustness of offline signature verification to different influencing factors [22]. The novel part is adding different levels of noise to signature images, simulating real bank checks. Baseline verification method follows from [3]. Local derivative pattern feature gives the best result of 15.35% EER with 10 references using GPDS-300 database which is a superset of GPDS-160 [23]. In case of adding the maximum level of noise level, EER reaches to 16.43%.

Ganapathi and Rethinaswamy present a person-dependent off-line signature verification using fuzzy techniques in image contrast enhancement, feature extraction and verification based on similarity measure [24]. First, experiments are conducted on the signature images where the features extracted using gray level intensity are characterized by interval-valued fuzzy sets and classified as genuine or forgery, using a similarity score. Then, signature images are contrast intensified using fuzzy sets / intuitionistic fuzzy sets and verified as above. Reported DER is 12.56% on CEDAR dataset [25] with 12 genuine signatures used as references per user.

Features: There are many different features that are used in the offline signature verification literature. For instance, in one of the earlier works, *local shape descriptors* are used as features [26]. A representation of handwritten signatures by *conics* (straight lines, ellipses and hyperboles) is presented by Bastos et al. [27]. This representation allows a simplification of the signature. However this simplification does not provide an enhanced verification success, instead it is used for the purpose of verification in the context of random forgeries, when forger doesn't imitate the original signature.

Shape matrices are studied in the context of offline signature verification by

Sabourin et al. [26]. First step is the evaluation of the centroid of the object under study. The second step lies in the evaluation of the main orientation of the pattern in the 2D space. In the case of handwritten signatures, the baseline of the signature is the natural choice for this class of patterns. These operations can be implemented with the evaluation of statistical moments. Consequently, invariance in translation and in orientation is obtained by this process. The third step is to locate the circumscribing circle of the pattern under study. Once the binary shape matrices are calculated, it is straightforward to measure the similarity between these matrices by the number of corresponding points. Several similarity measures are compared. A best DER of 0.84% on a private database with random forgeries for testing is reported.

Later, *local correspondence* between a model and a query signature is used to compare a set of geometric properties [28]. *Interior stroke distributions* in polar and Cartesian coordinates are used in the work by Ferrer et al. [29]. In the work by Nguyen et al. [30], *enhanced modified direction feature* (MDF) is utilized. Later, Nguyen et al. use basic global features extracted from the whole signature (e.g. width/height ratio or pixel density) [21].

Radon transform is used to extract features to feed to a HMM [31]. Later, another offline signature verification system that utilizes Radon transform is introduced by Panton [32]. HMMs are trained from features extracted from local regions of the signature (local features), as well as from the signature as a whole (global features). To achieve this, each signature is zoned into a number of overlapping circular retinas, from which said features are extracted by implementing the discrete Radon transform. A global retina, that encompasses the entire signature, is also considered.

A fuzzy modeling that employs the *Takagi-Sugeno* (TS) model is proposed by Hanmandlu et al. [33]. Distance distributions and angle distributions are extracted from image partitions. Because the same feature may exhibit variation in different samples, rise to a fuzzy set is given. The features are fuzzified by an exponential membership function involved in the TS model, which is modified to include structural parameters. The structural parameters are devised to take account of possible variations due to handwriting styles and to reflect moods. The membership func-

tions constitute weights in the TS model. The optimization of the output of the TS model with respect to the structural parameters yields the solution for the parameters. Two TS models are derived by considering a rule for each input feature in the first formulation (multiple rules) and by considering a single rule for all input features in the second formulation. It is reported that TS model with multiple rules is better than TS model with single rule for detecting forgeries.

Left-to-Right HMMs (LR-HMM) are utilized in order to extend those models to the field of static or off-line signature processing using results provided by image connectivity analysis in the work by Igarza et al. [34]. The *chain encoding* of perimeter points for each blob obtained by this analysis is an ordered set of points in the space, clockwise around the perimeter of the blob. Two models are generated depending on the way the blobs obtained from the connectivity analysis are ordered. In the first one, blobs are ordered according to their perimeter length. In the second proposal, blobs are ordered in their natural reading order, i.e. from the top to the bottom and left to right. Finally, two LR-HMM models are trained using the (x,y) coordinates of the chain codes obtained by the two mentioned techniques and a set of geometrical local features obtained from them such as polar coordinates referred to the center of ink, local radii, segment lengths and local tangent angle. MCYT baseline corpus is used for experimentation where a best of 27.58% EER is reported with skilled forgeries. In a more recent work by Bharathi and Shekar, the four-directional chain code histogram of each grid on the contour of the signature image is extracted [35]. Subsequently, the SVM classifier is used as the verification tool. GPDS-100 is considered to test the system, where 11.4% DER is reported using 12 genuine references per user.

Contour features are extracted in the work by Gilperez et al. [36]. Considered features are: Contour-Direction probability distribution function (PDF) representing the histogram of angles, Contour-Hinge PDF (2 contour fragments attached at a common end pixel is considered and joint probability distribution of the orientations of the two sides is computed), Direction Co-Occurrence PDFs (combination of contour-angles occurring at the ends of run-lengths on the background are used), Run-Length PDFs (regions enclosed inside the letters and strokes and also the empty spaces between them are captured both vertically and horizontally). To compare

the PDFs of a query and a reference, χ^2 metric is used. Feature level combination is also investigated. Mean value of the Hamming distances due to the individual features is used as the similarity metric, in that case. Best working feature among the explained PDFs is Contour-Hinge PDF, individually working at 10.18% EER with 5 genuine signatures as reference set per user, utilizing the MCYT corpus. No feature level combination is reported to perform better than the individual Contour-Hinge PDF.

Later by Larkins and Mayo, features such as *gradient direction* and *equimass spatial pyramids* are extracted before binarizing the feature vectors by thresholding [37]. Adaptive feature thresholding (AFT) is proposed as a method of person-dependent off-line signature verification. AFT enhances how a simple image feature of a signature is converted to a binary feature vector by improving its representation in relation to the training signatures. The similarity between signatures is then easily computed from their corresponding binary feature vectors. This method is tested on GPDS-39 dataset and 14.01% DER is reported with 12 references.

Local interest points, which correspond to local maxima in a scale-space representation of a signature, are detected in the publication by Solar et al. [38]. The descriptors that characterize local neighborhood around corresponding interest points, are calculated using the scale invariant feature transform (SIFT). The correspondence between descriptors of reference and query signatures is established using wide baseline methodology, while the final decision is performed using a Bayes classifier. The system performance is assessed using the GPDS-160 signature dataset, where 15.3% DER is reported. However, a full skilled forgery test is not performed, just a small subset of all skilled forgeries for testing is used. A novel signature stability analysis based on signatures' local and part-based features is presented by Malik et al. [39]. Speeded up local features (SURF) are used for local analysis which give various clues about the potential areas from whom the features should be exclusively considered while performing signature verification. Locally stable SURF gives 15% EER on 4NSigComp2010 dataset which is the best result reported so far.

High pressure points in polar coordinates are adapted to the problem by Vargas et al. [40]. Features representing information about pressure distribution from a static image of a handwritten signature are analyzed for an offline signature ver-

ification system by Vargas et al. [41]. From gray-scale images, its histogram is calculated and used as spectrum for calculation of pseudo-cepstral coefficients. The unique minimum-phase sequence is estimated and used as feature vector for signature verification. The optimal number of pseudo-coefficients is estimated for best system performance. Experiments are carried out using gray-level GPDS-100. The robustness of the analyzed system for simple forgeries is tested with 12 genuine and 12 skilled forgery signatures as reference set per user to report 6.20% EER.

Stroke gray-level variations are measured by Vargas et al. by means of *wavelet analysis* and *statistical texture features* [42]. This method begins with a proposed background removal. Then wavelet analysis allows to estimate and alleviate the global influence of ink-type and finally, properties of the co-occurrence matrix are used as features representing individual characteristics at local level. Results are provided with gray-level GPDS-100 database (gray-level version of a simpler subset of GPDS-160). Utilizing 5 random genuine samples as reference gives an EER of 14.22%.

Histogram of oriented gradients (HOG) features are used by Zhang for offline signature verification problem [43]. A local shape descriptor pyramid histogram of oriented gradients (PHOGs), which represents local shape of an image by a histogram of edge orientations computed for each image sub-region, quantized into a number of bins is applied. Each bin in the PHOG histogram represents the number of edges that have orientations within a certain angular range. An early version of GPDS database, GPDS-39 is used for experimentation. For each subject; 19 genuine signatures and 24 skilled forgeries are picked out for training, leaving 5 genuine signatures and 6 skilled forgeries for testing. For the above-stated configuration, 3.63% DER is reported.

Recently, *graphometric features* started to draw attention. A graphometric feature set that considers the curvature of the most important segments of the signature is introduced by Bertolini et al. [44]. Shape of the signature is simulated by using Bezier curves and then features are extracted from these curves. Parodi et al. propose an approach [45] to make some basic set of features invariant to rotation, with the help of Discrete Fourier Transform (DFT). Considered features are static graphometric features such as the number of pen pixels inside a circular sector over

the area of the circular sector. Same features are calculated inside rotated versions of circular sectors, followed by DFT. It is justified that the feature set obtained is invariant to rotation. Random 30 subjects of GPDS-160 are dedicated for parameter optimization. Remaining 130 subjects are trained with 13 genuine signatures and 129 random forgeries per writer. Each subject is tested with simple and skilled forgeries, where simple forgery test set is not detailed. Without any rotation, 4.21% EER is reported.

Guest and Miguel-Hurtado apply a fingerprint matching method (*fingercode*) to offline signature verification [46]. Three other methods (geometric centroids, global and local features, geometric features) are also implemented for comparison. Fingercode methods give 32.45% and 30.78% EER with 5 and 10 references from each user on the GPDS-300 dataset. Majority voting classifier combination of the 4 methods achieves 12.59% and 11.22% EER with 5 and 10 references from each user.

Statistical texture features are successfully applied to offline signature verification. Complex features based on *local binary patterns* (LBP) (so called pseudo-dynamic features) to perform statistical texture analysis are introduced by Vargas et al. [3]. To extract second order statistical texture features from the image, another feature called the *gray level co-occurrence matrix* (GLCM) method is utilized. Best combination with 10 genuines used as reference set with gray-level GPDS-100 database gives an EER of 9.02%. Ferrer et al. use local derivative pattern feature, giving the best result of 15.35% EER with 10 references using GPDS-300 [22]. Hu and Chen use pseudo-dynamic features based on gray level: LBP, GLCM and HOG [47]. Wajid and Bin Mansoor also use LBP as feature [48]. Ganapathi and Rethinaswamy present a person-dependent off-line signature verification [24]. Features extracted are gray level intensity characterized by interval-valued fuzzy sets. Reported DER is 12.56% on CEDAR dataset with 12 genuine signatures used as references per user.

Deep learning is a research area that has growing interest. A deep learning model for off-line handwritten signature recognition which is able to extract high-level representations is presented by Ribeiro et al. [49]. A deep neural network is utilized to extract a high level representation of the signature images. However no result is published for deep learning part, published results instead make use

of conventional features (MDF, width, height) and conventional classifiers (SVM). Khalajzadeh et al. [50] propose an offline signature verification scheme based on Convolutional Neural Network (CNN - [51]). CNN is utilized for feature extraction without prior knowledge on the data. The classification task is performed by multilayer perceptron network (MLP). Proposed method is intended to be robust to signature location changes and scale variations. A private database of 176 signatures from 22 subjects is used for experimentation. No detail is provided about the experimental setup, mean squared test error is reported to be lower than 0.1%.

Partially ordered grid features are used to measure signatures' structural characteristics by Zois et al. [52]. Thirty-two binary symbols are delineated within the five-by-five pixel window and considered to be the alphabet of a probabilistic source. The whole set is organized into subsets of four symbols each. The new arrangement is used to detect the presence of simple or compound symbols in the signature image. The utilization of the partially ordered set (poset) notion arranges the binary feature extraction masks into first order chains. This way a first order probabilistic description of the signatures structure that is characteristic of the motoric signature generating process is supposed to be created. First order searching strategy is limited to pixels neighbors having their grids centered to a predetermined Chebyshev distance of two. SVM is used for verification. Using 5 genuine and 5 skilled forgeries for training leads to an EER of 6.64% while using 12 genuine and 12 skilled forgeries for training leads to an EER of 3.21% on GPDS-300 database.

Matching the template and query: It is of common interest of many works in the literature to match the template and query by using the extracted features or raw signature images. Abuhaiba presents a simple and effective signature verification method that depends only on the raw binary pixel intensities and avoids using complex sets of features [53]. The method looks at the signature verification problem as a graph matching problem. The method is tested using genuine and skilled forgery signatures produced by five subjects. An EER of 26.7% is achieved for skilled forgeries. In a study by Shanker and Rajagopalan, vertical projection features are used as features fed into a DTW algorithm with some modifications to incorporate a stability factor to increase the performance of the DTW algorithm [54]. The system gives a DER of 22.5% on skilled forgery test using a private database.

There exists plenty of works to adapt snakes-related algorithms to offline signature verification. Vélez et al. publish a short review and comparison of these methods [55]. Considered methods are shape-memory snakes and parallel segment matching. Snake features that are used for classification are coincidence, distance and energy. In parallel segment matching, at the end of iterative elastic adjustment, the mean Euclidean distance between the corresponding matched segments of the two compared signatures is computed. This value is compared to an experimental threshold (which is computed using the three training signatures) to decide whether the test signature is authentic or it is a forgery. Experimental results show that the shape-memory snakes clearly outperform to the parallel segment matching approach on the same signature dataset (9% EER compared to 24% EER respectively, on a private database).

Offline signature verification by affine registration of genuine and forgery signatures' 2D point sets is proposed by Tian and Lv [56]. Each point in genuine and forgery signatures is considered as a complex number and from each point set, a polynomial with complex coefficients can be computed whose roots are the points in the given point set. Then a verification function is achieved based on a difference between the points which can be determined by an unknown rotation. In order to archive the rotation, a two-step algorithm is employed. First the affine registration problem is reduced to a rigid registration problem, and the unknown rotation is then computed using the coefficients of these polynomials. System performance is measured with GPDS-39 database and 12 genuine references are used per subject. Reported result is 13.08% DER.

Classification: There are many different classifiers that have been applied to offline signature verification so far. Bayes classifier is used by Solar et al. [38]. K-nearest neighbor (KNN) classifier is one of the simplest choices and used for offline signature verification [26]. A comparison of probabilistic neural networks (PNN) and KNN is done by Vargas et al. [40]. Genuine and skilled forgery signatures of each subject are divided into two equal parts; making 12 genuine and 12 skilled forgery training signatures and the same amount of test signatures. Best KNN result is 12.62% DER and best PNN result is 12.33% DER on gray-level GPDS-160 database.

Neural networks are used especially in former works. Two approaches are used by Xiao and Leedham to exploit information related to stable parts of signatures (the parts that do not show much variation across the signatures of a user) [57]. The first approach is to train a neural network classifier with artificial forgeries generated by removing stable components from genuine signatures, so that the classifier detects changes in these stable components when verifying signatures. The other is to force the neural network classifier to pay special attention to local stable parts of signatures by weighting their corresponding node responses through a feedback mechanism. Neural networks are also used by Nguyen et al. [30] in a later work for comparison.

HMM is one of the popular choices for offline signature verification [31,32,34,58, 59]. Coetzer and Sabourin propose a system that is semi-automatic and combines computer verification systems with manual human verification [60]. This combined system is shown to perform better than humans or a machine for almost all operating costs. HMM classifier outperforms most of the individual human verifiers (21/23). In spite of this result, it is also shown that the maximum attainable combined classifiers outperform the HMM classifier, and the most proficient human classifiers, for most operating costs.

SVM is the most common classifier in the context of offline signature verification. A comparison of SVM and HMM classifiers in the context of the off-line signature verification is reported by Justino et al. [58], where a private database of 100 subjects is utilized to compare the classifiers. Both of the classifiers are trained using signatures of the first 40 subjects, and tested using signatures of the remaining individuals. According to the reported results, SVM is found to be superior to the HMM classifier. HMM, SVM and simply the Euclidean distance are compared by Ferrer et al. [29]. The GPDS-160 database is used to evaluate the method. Three skilled forgery signatures from each subject are used for training purposes, which may not be realistic since it requires knowledge of existing forgeries for each user. Authors report performance results based on DER, which is the average of FAR and FRR. When 12 genuine signatures are used as reference, remaining 12 genuine and 27 skilled forgery signatures are used for testing each person; HMM gives 13.35% DER, SVM with radial basis function (RBF) kernel gives 14.27% DER and

Euclidean distance metric gives 15.94% DER. In the work by Nguyen et al. [30], MDF is utilized with artificial neural network (ANN) and SVM used as classifiers. 12 genuine signatures are used for training and 100 writers are randomly selected to provide 400 random forgeries as negative examples. For testing, authors use a mix of random and skilled forgeries where the remaining 12 genuine signatures are used together with 59 random forgeries from the remaining 59 writers and 15 targeted (skilled) forgery signatures of that specific writer. They obtain 20.07% DER with SVM on GPDS-160 database. Usually, the bi-class SVMs (B-SVM) are used for separating between genuine and forged signatures as also done in this thesis. However, in practice, only genuine signatures are available for training, other than random forgeries. Guerbai et al. use one-class SVM (OC-SVM) for handwritten signature verifications [61]. Experimental results conducted on CEDAR database show the effective use of the one-class SVM (4.39% DER) compared to the biclass SVM (14.46% DER). There are other recent works using the SVM classifier as the verification tool [35, 47].

Wajid and Bin Mansoor investigate the performance of seven different classifiers with LBP as feature [48]. Classifiers are Least Squares-SVM (LS-SVM), SVM, Distance Likelihood Ratio Test (DLRT), ANN, Fisher's linear discriminant, Logistic Discriminant, Naive Bayes. Experimental findings depict that LS-SVM performs the best among the seven classifiers.

User-independent verification: Natural way to train the classifiers is user-based. However, user-independent classifier training is another possibility. A global offline signature verification system is proposed by Santos et al. [62]. Feature difference vectors are calculated via each reference. Majority decision calculated via decision of each reference's difference between the query is taken as the final decision.

A hybrid writer-independent (WI) and writer-dependent (WD) offline signature verification system is proposed by Eskander et al. [63]. A global classifier is designed using a development database, prior to enrolling users to the system. When a user is enrolled to the system, a WI classifier is used to verify his queries. During operation, user samples are collected and adapt the WI classifier to his signatures. Once adapted, the resulting WD classifier replaces the WI classifier for this user. Suitable switching point between the WI and WD modes is identified by the number of

training samples that produce WD classifiers with higher accuracy than the global WI classifier. Classification method is similar to the one proposed in this thesis, however our system can work without any user-based (WD) classifier on demand or if enough user specific references are provided, they can be utilized with the help of score level fusion. Our global classifiers take a bit long to train but can work alone without further training as stated. GPDS-300 database is used to evaluate the system in [63] where 140 users are devoted as the development set and 160 users are devoted for training; exactly the same as our configuration. With WD classifier, 22.71% DER is obtained when 12 genuine signatures are kept as reference and skilled forgeries are utilized as negative test samples. Under the same configuration with WI classifier, 26.73% DER is obtained.

An offline signature verification system using two different classifier training approaches is proposed by Hu and Chen [47]. In the first mode, each SVM is trained with the feature vectors obtained from the reference signatures of the corresponding user and those random forgeries for each signer while the global Adaboost classifier is trained using genuine and random forgery signatures of signers that are excluded from the test set. Global and writer-dependent classifiers are used separately. Combination of all features for writer-dependent SVMs results in 7.66% EER for gray-level $GPDS_{random}150$ with 10 references. Combination of all features for writer-independent Adaboost results in 9.94% EER for gray-level $GPDS_{random}100$ with 10 references. Here, $GPDS_{random}150$ denotes randomly selected 150 subjects of gray-level GPDS-300 and $GPDS_{random}100$ denotes randomly selected 100 subjects of gray-level GPDS-300.

Classifier combination: Classifier combination helps further improvements as in many other fields. A multi-hypothesis approach and classifier fusion is utilized by Panton [32]. Each base classifier is constructed from a HMM that is trained from local features, as well as from global features. An ensemble of classifiers based on graphometric features is utilized by Bertolini et al. [44] to improve the reliability of the classification. The ensemble is built using a standard genetic algorithm and different fitness functions were assessed to drive the search. Guest and Miguel-Hurtado apply majority voting classifier combination of 4 different features (fingercode, geometric centroids, global and local features, geometric features) [46]. They achieve

12.59% and 11.22% EER with 5 and 10 references from each user, compared to single Fingercode method giving 32.45% and 30.78% EERs, respectively. Experiments are carried out on the GPDS-300 dataset.

Hybrid generative discriminative ensembles of classifiers (EoCs) are proposed by Batista et al. to design an offline signature verification system from few references, where the classifier selection process is performed dynamically [59]. To design the generative stage, multiple discrete left-to-right HMMs are trained using a different number of states and codebook sizes, allowing the system to learn signatures at different levels of perception. To design the discriminative stage, HMM likelihoods are measured for each training signature, and assembled into feature vectors that are used to train a diversified pool of two-class classifiers through a specialized Random Subspace Method. During verification, a new dynamic selection strategy based on the K-nearest-oracles (KNORA) algorithm and on Output Profiles selects the most accurate EoCs to classify a given input signature. GPDS-160 database is used to evaluate the system and 16.81% EER is reported using 12 references per user.

User-based score normalization: Score normalization is reported to improve the system performance in many biometric modalities. In the work by Ferrer et al. [29] to find user-based thresholds, three skilled forgery signatures from each subject are used, which may not be realistic since it requires knowledge of existing forgeries for each user. A score normalization scheme is also applied to make individual user's scores consistent with global system EER threshold by Panton [32].

Signature recognition: Recognition is not a common practice in the context of offline signatures. Özgündüz et al. explored the recognition accuracy when only genuine samples are input to the system [64]. Basic features such as area or mask features are used to report a recognition accuracy of 95% with SVM as the classifier.

Online signatures for enrollment: Yu et al. make use of online handwriting for enrollment, instead of handwritten images [65]. Online reference signatures enable robust recovery of the writing trajectory from an input offline signature and thus allow effective shape matching between reference and query signatures. In addition, several techniques to improve the performance of the signature verification system is proposed: Trajectory is recovered within the framework of Conditional Random Fields; a new shape descriptor called online context is introduced for align-

ing signatures; a verification criterion which combines the duration and amplitude variances of handwriting is developed. Training is done as in online signature verification, however test samples are converted to static images as in offline signature verification for evaluation. Results are compared with purely online and purely offline systems. They use SVC 2004 database [66] for experimentation. EER is 7.3% and 7.4% on set 1 and set 2. Best offline results that they reference are 23.3% and 22.0% EER on the same sets. Best online results are 5.8% and 4.6% EER on the same sets.

Biometric template security: Biometric template security is a well studied topic, however it has just started to draw attention in offline signature verification area. Impact of watermarking attacks on the performance of offline signature verification is assessed in the context of intelligent bio-watermarking systems by Rabil et al. [67]. Extended Shadow Code (ESC) features are extracted from digitized offline signatures, collected into feature vectors, and discretized into binary watermarks prior to being embedded into high resolution grayscale face image. The impact on biometric verification performance of quantization and different intensities of attacks are considered. The impact of using only certain areas of face images of higher texture region of interest (ROI) for embedding the watermark is observed.

A Fuzzy Vault (FV) system based on the offline signature images is proposed by Eskander et al. [68]. A two-step boosting feature selection (BFS) technique is proposed for selecting a compact and discriminant user-specific feature representation from a large number of feature extractions. Representation variability is modeled by employing the BFS in a dissimilarity representation space, and it is considered for matching the unlocking and locking points during FV decoding. The limited discriminative power of FVs is alleviated by using an additional password, so that the FAR is reduced without significantly affecting the FRR. Enhancing system accuracy comes with the expense of the user inconvenience. Experiments are carried out on a Brazilian database and skilled forgery tests ends up with 15.48% DER using 15 signatures templates.

A novel user-convenient approach is proposed by Eskander et al. [69] for enhancing the accuracy of signature-based biometric cryptosystems. Since signature verification (SV) systems designed in the original feature space have demonstrated

higher discriminative power to detect impostors, they can be used to improve the FV systems. Instead of using an additional password, the same signature sample is processed by a SV classifier before triggering the FV decoders. Using this cascaded approach, the high FAR of FV decoders is alleviated by the higher capacity of SV classifiers to detect impostors. With the cascaded SV-FV approach, 15.48% DER is reduced to 11.13%.

Databases: Currently, there are many public databases for common use; including GPDS (Grupo de Procesado Digital de Senales) [23], MCYT (Ministerio de Ciencia Y Tecnologia) [19], CEDAR (Center of Excellence for Document Analysis and Recognition) [25], SVC-2004 (Signature verification competition) [66], Caltech [70], HIT-MW Chinese signature database [71], PUCPR Brazilian database (Pontificia Universidade Catolica do Parana) [72].

Current state of the art among the works where no skilled forgery of a user is utilized in training phase is reported to be 4.21% EER [45]. The work considers 13 genuine signatures as reference per user and utilizes a random 130 subjects of GPDS dataset for experimentation. Test set includes skilled forgeries and simple forgeries which is not detailed. In Table 5.9, we give the summary results for the systems utilizing GPDS dataset. Performance results are summarized in the form of DER to be compatible with the previous results.

To measure the improvement with a particular contribution, we utilize a baseline system that is defined in detail in Section 5.2.1. This system will be referred to as **baseline** within the scope of this thesis. We determine whether to use a specific method in our final system according to the reported results with the baseline.

Previous versions of our signature verification system won several competitions. In 4NSigComp2010 [73], we won task one, where 90 forgery, 3 genuine, 7 disguised for test were existed. Without counting the disguised, we obtained 86.02% accuracy. We won Chinese signatures category with 80.04% accuracy in 4NSigComp2011 [74]. Our system won the 4NSigComp2012 [75], category without disguised forgeries. Our system was the winner of all offline categories in SigWiComp2013 [76].

1.3 Contributions

Our main contribution in this thesis is a comprehensive treatment of all aspects of offline signature verification, resulting on a state-of-art verification system that has achieved first place in several signature verification competitions. Aspects that contribute to the success of this system are listed below.

1. We propose new preprocessing techniques to alleviate the problem of large variations in embellishments (strokes that often begin or end a signature, changing the shape drastically) and pen thickness: methods such as removal of outlier signature parts end up with a loss of information, but they are well suited for handling irrelevant variations among genuine signatures.

2. We developed a technique to align the signature images to references automatically. Registration is applied on the training stage of global classifier such that each query signature of each user in the training set is aligned to each reference of that user. Signature alignment brings more than 2% improvement on average.

3. We utilize complementary features such as HOG, LBP, and SIFT in order to achieve high accuracies. Furthermore, we improve upon the basic feature methodologies by novel adaptations in each case. i) we use coarse-to-fine grids for capturing a spectrum of global to highly local features (signature's invariant features). ii) We select best LBP templates according to term frequencies and combine similar LBP template histogram bins to obtain a dense histogram. Our LBP application is one of our major contributions that brings an error rate lower than the state of the art in the same domain of offline signature verification. iii) For SIFT, we use a novel matching algorithm that seeks more than one global transformation, in order to allow different transformations in different parts of a signature.

4. We incorporate user-dependent and user-independent verification concurrently. We do this by training the global classifiers once, then training user dependent classifiers for each individual with limited number of reference signatures.

We then apply a score level fusion to combine classifiers with complementary feature types, where the weights are learnt from a separate validation set.

5. We present experiments on the effects of user-dependent score normalization. We develop a novel score normalization method that performs better than known techniques, without using any skilled forgeries in training.

1.4 Outline

The rest of the thesis is organized as follows:

In Chapter 2, importance of preprocessing and our preprocessing stage is described in detail. Image preprocessing is an inevitable stage in nearly all problems dealing with digital images as stated in Section 2.1. We explain our preprocessing methodology in Section 2.2.

Feature extraction is an important key of this work, which is explained in detail in Chapter 3. Common features that have been applied to offline signature verification problem are shortly described in Section 3.1. Section 3.2 covers the coordinate systems (Cartesian and polar coordinates) and fixed number of overlapping grids which localize the features. Section 3.3 introduces the HOG features, Section 3.4 introduces the LBP features and Section 3.5 introduces the SIFT features that we use. Especially LBP and SIFT features are modified and improved to fit well into our domain of offline signature verification.

In Chapter 4, we explain our classification method that outputs the final verification decision. Global classifier is explained in Section 4.1, which is followed by user-based classifier in Section 4.2. At the end, we explore user-based score normalization in Section 4.3. Although it improves the performance of systems such as speaker identification, even more complicated techniques that we implement are not successful to come up with a relationship between reference images of a user and the ideal score shift of the corresponding user. In Section 4.4, our classifier combination approach is described.

In Chapter 5 experimental results are presented. The dataset that is used to obtain a performance measure of our system is explained in Section 5.1. Different test configurations are introduced in Section 5.2. In Section 5.3, error rates of

partial features and classifiers are given with the error rates of full system in detail. A comparison with other works in literature using similar test configuration is also provided in the same section. Running times of different modules of the system are shown in Section 5.4.

Finally in Chapter 6, conclusions and proposed future work are reported.

Chapter 2

Preprocessing

2.1 Motivation

Signature images have variations in terms of pen thickness, embellishments found in strokes, translation or relative position of strokes, rotation, scaling even within the genuine signatures of the same subject. Because a verification system takes into account only static signature images, signature images should be normalized well before they are further processed. Sample genuine (first three columns) and their corresponding skilled forgery (last column) signatures from GPDS dataset [23] are shown in Figure 2.1.

2.2 Method

Our first step is to remove connected components consisting of a few pixels (such as less than 20) that are not expected to happen in all signatures of a specific user. This kind of connected components rather contribute as noise and does not provide any useful information. Next, a **bounding box** should be established which provides a rectangular workspace. Initially, bounding box is determined as the rectangular box with minimum and maximum horizontal and vertical coordinates of signature pixels. Example results of first two preprocessing steps are shown in Figure 2.2 along with the original signature. Initial bounding box is subject to further modifications, as explained in following steps.

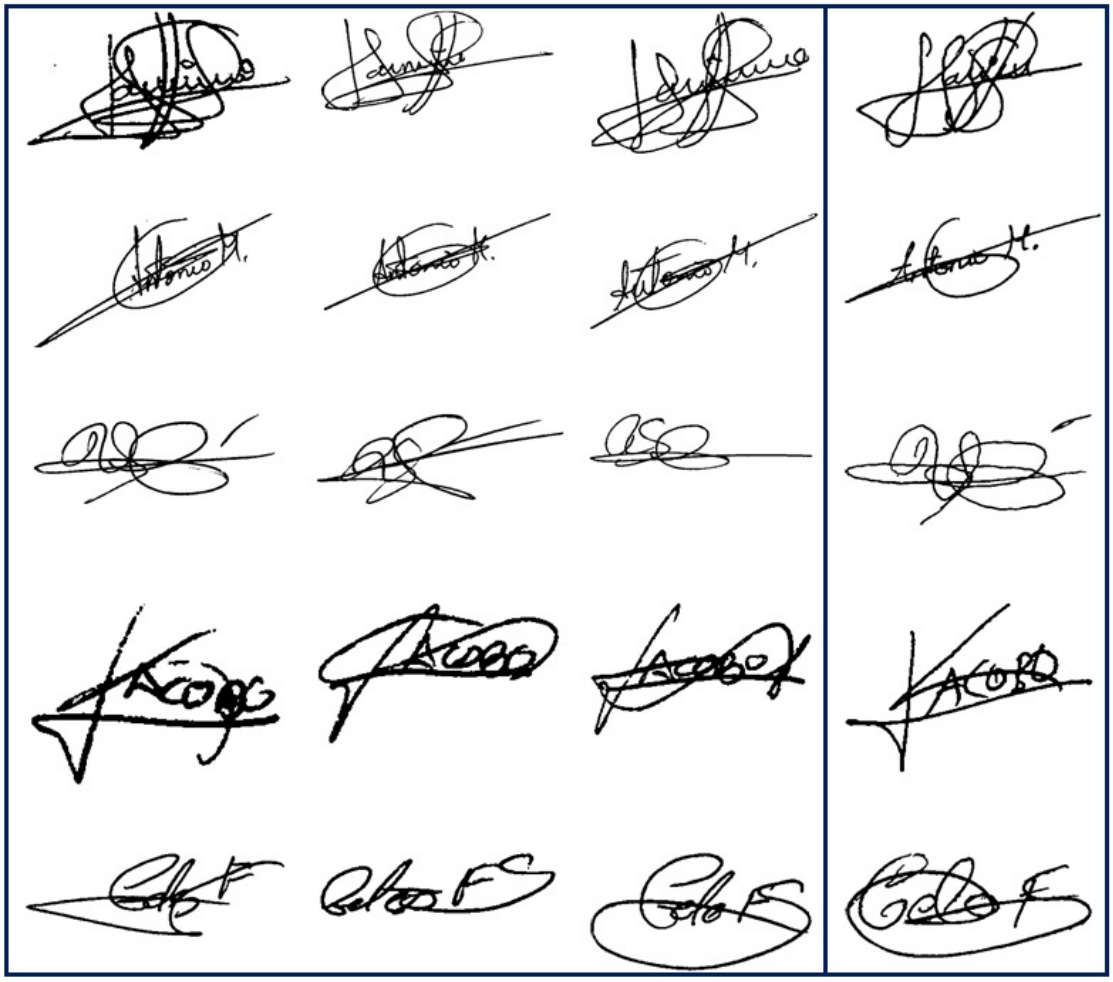
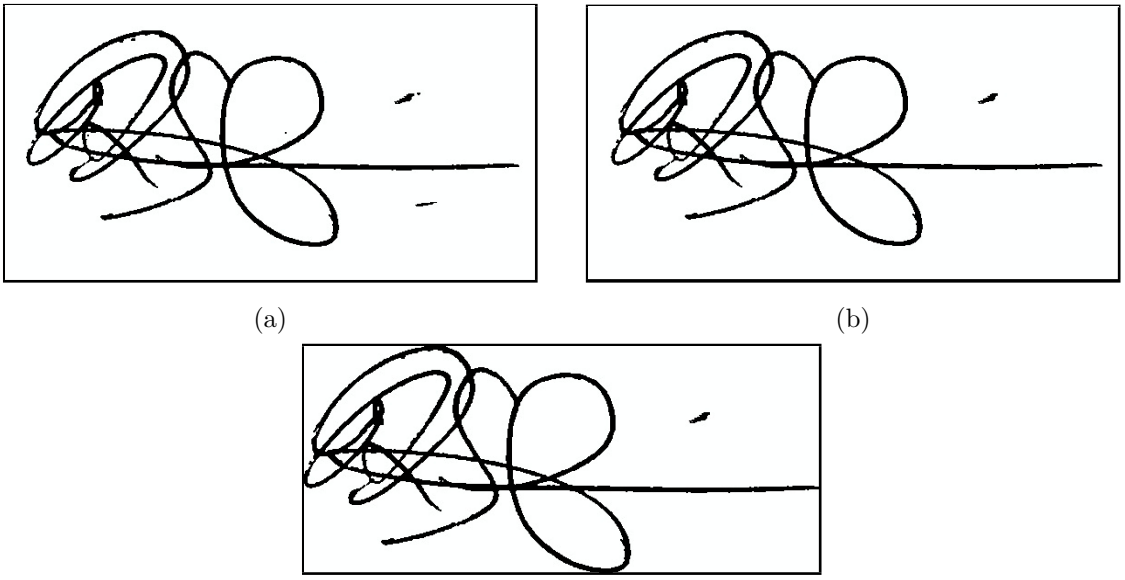


Figure 2.1: Sample genuine (first three columns) and their corresponding skilled forgery (last column) signatures from GPDS-160 database.



(a) (b)

(c)

Figure 2.2: Basic preprocessing steps: (a) Original image, (b) Small connected components removed, (c) Min-max bounding box.

Our system is intended to be robust to global shape variations that are commonly induced by variations in embellishment that are produced by fast ballistic movements and are floating inside the signature's overall pattern. Suppose that some kind of variation in embellishment is common to all signatures of a user. Then just removing these strokes should not effect the verification performance in ideal case. It could also be possible that some variation in embellishment exists in some of the genuine signatures. Then one of these two approaches should work: Remove this stroke from all claimed signatures of that user, or extend all claimed signatures of that user with zero padding. To come up with a simple and global normalization of variations in embellishment, we handle such variations by modifying the bounding box.

Strokes that are far away from image centroid are cut by cropping the bounding box. This was done using a distance threshold which is derived from the standard deviation of the trajectory points' coordinates ($\approx 3 \times \sigma$). This normalization is supposed to help compensate for translation variations which will prevent grids from fitting in the same signature locations. A signature with initial min-max bounding box and with narrowed bounding box are shown in Figure 2.3.

(a) (b)

Figure 2.3: Further preprocessing steps: (a) Min-max bounding box, (b) Narrowed bounding box.

To compensate for pen thickness variations, we find the signature contour and use it instead of the signature image itself. Skeletonisation is another possibility for the same purpose, however it looses some details of the signatures, like user specific shapes as can be seen in Figure 2.4. This step is found to be useful in feature types which use gradient information (HOG), experimentally. However we skip this step for the feature types which intensely use texture information, namely LBP and SIFT.

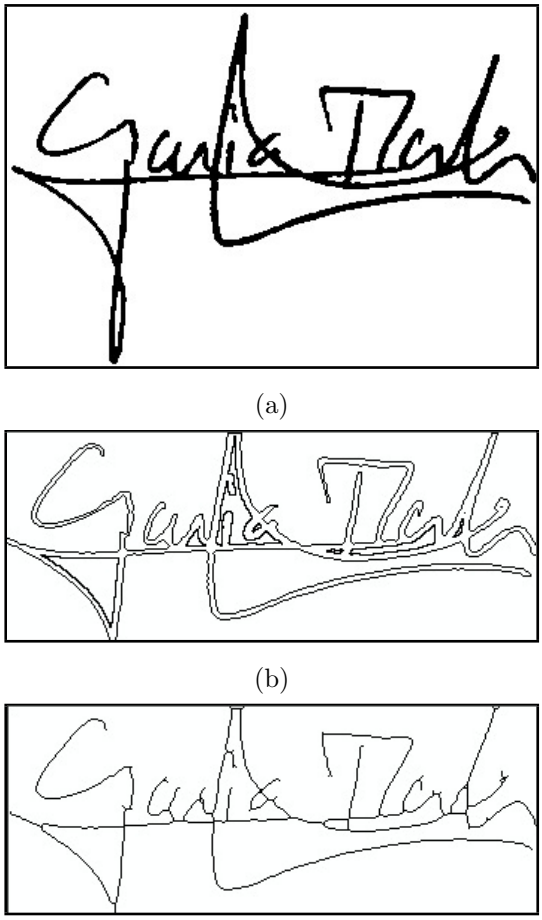


Figure 2.4: Preprocessing (a) Original signature (b) Contour image (c) Skeleton image.

One of the significant difficulties in comparing offline signatures is the lack of registration between the signatures. There are no robust reference points in signatures, and individual strokes at the beginning or end of signatures change the appearance easily. Rotation normalization may be handled by utilizing image moments. For example Kalera et al. perform rotation normalization by rotating the signature curve until the axis of least inertia coincides with the horizontal axis [77].

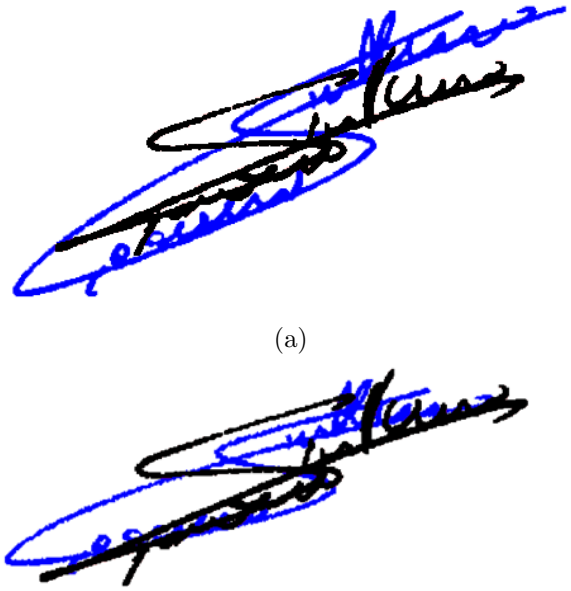
Merits of both global and local alignment methods are incorporated by You et al. [78]. Two signature patterns are globally registered using weak affine transformation. Correspondences of feature points between two signature patterns are determined by applying an elastic local alignment algorithm. Similarity is measured as the mean square of sum Euclidean distances of all found corresponding feature points based on a match list.

Offline signature verification may benefit from normalization steps to obtain global rotation, scale and translation invariance, since signing conditions may significantly change size, orientation and location of the signature in a document. We initially normalize the effects of translation by adding empty rows and columns to signature image, making the image centroid the same as image center. We describe our method to normalize the effects of rotation, scaling and fine translation below.

Each query signature image Q of a training user is aligned to each reference R^i of that user with best scaling (σ), rotation (θ) and translation (δ) parameters, obtaining $Q^i_{\sigma,\theta,\delta}$. Best parameters are determined as the ones which maximize the similarity or minimize the distance of query to reference image:

$$argmin_{\sigma,\theta,\delta}\{\|Q^i_{\sigma,\theta,\delta} - R^i\|\}. \tag{2.1}$$

As similarity metric, ℓ_1-norm of the Euclidean distance between LBP features is used. An example reference, query and aligned query is shown in Figure 2.5. For faster alignment, we apply all possible transformations to reference R^i once for an enrolled user. We find the best parameters indicated in equation 2.1 exhaustively and apply the inverse transformation to Q using $1/\sigma$, $-\theta$ and $-\delta$. We use a small interval to search for best transformation: -2.5 to +2.5 degrees for θ, 0.8 to 1.2 for σ, -10 to +10 pixels for δ. These intervals currently seem to be enough as there are no significant alignment differences in the database. Larger intervals naturally increase the cost of search and should be handled by more sophisticated methods such as iterative closest point (ICP) or random sample consensus (RANSAC), possibly utilizing the SIFT matching. After alignment, feature vector of aligned query Q is extracted.



(a)

(b)

Figure 2.5: Alignment example: a) not aligned and b) aligned reference and query.

Signature alignment is implemented only on the training phase of global classifier, so as to obtain better aligned features from the reference signatures. It is experimentally found that alignment of queries during testing does not improve overall performance. This is due to the fact that although some genuine queries get higher scores when aligned, some forgery queries also get higher scores when they are aligned with references. In contrast, during training, we know the label of aligned signatures (genuine/forgery) and this process indeed improves the overall performance. The effect of alignment improves the verification accuracy by decreasing EER of individual global classifiers as reported in Table 5.3. We do not apply alignment in training of user-based classifiers, because the only available information specific to a user is some limited number of reference images in user-based classifiers (Section 4.2).

Chapter 3

Feature Extraction

3.1 Overview

Feature extraction step reduces the dimension of original signature images while preserving and extracting the important information encoded in the image. A carefully selected set of features will transform the images so that it becomes easier to distinguish between genuine and forgery classes. Weaker features will increase the load on the classifier. In this section, common features that have been used in offline signature verification problem are summarized.

Local shape descriptors: Local shape descriptors (LSD) cover a wide variety of global descriptors including the shape context and high pressure points. LSD provide surface correspondence and feature detection functionalities. Global descriptors are localized by local segments of the image [26]. Local granulometric size distributions are used as a local shape descriptor by means of morphological operators [40].

Radon transform: Radon transform is the integral transform consisting of the integral of a function over straight lines. It is widely applicable to tomography to create an image from the projection data associated with cross-sectional scans of an object. It is closely related with Hough transform which is the most popular technique for curve detection. Advantage of Radon transform over Hough transform is the whole mathematical basis. It is applied to offline signature verification [31,32].

Contourlet transform: Contourlet transform as introduced by Do and Vetterli [79] is an efficient tool for capturing smooth contours. It has five significant features:

Multiresolution, localization, critical sampling, directionality and anisotropy. It is a double filter bank: Laplacian Pyramid (LP) is followed by a Directional Filter Bank (DFB). It is also named pyramidal directional filter bank (PDFB). LP at each level decomposes input image into downsampled lowpass sub-band (coarse image) and one bandpass sub-band. DFB is then applied to bandpass sub-band. By repeating this scheme iteratively on the coarse image resulted from LP at each level, a fine to coarse representation of the input image is obtained. Contourlet transform is applied to offline signature verification problem by Pourshahabi et al. [80]. Reported EER values are 14% for a private Persian dataset and 23% for a private English dataset, with skilled forgeries.

Wavelet transform: Wavelet transformation is one of the popular candidates of the time-frequency transformations. The discrete wavelet transform is computationally less complex (O(N) time) as compared to the similar fast Fourier transform (O(N logN) time). Stroke gray-level variations are measured by Vargas et al. by means of wavelet analysis and statistical texture features [42]. Wavelet analysis allows to estimate and alleviate the global influence of ink-type.

Graphometric features: Graphometric features are intrinsic properties from an individual handwriting style, which may be employed by forensic experts during handwriting or signature recognition. These include curvature and pressure among others [81]. A graphometric feature set that considers the curvature of the most important segments of the signature is introduced by Bertolini et al. [44]. Shape of the signature is simulated by using Bezier curves and then features are extracted from these curves. Parodi et al. propose an approach [45] that consider static graphometric features such as the number of pen pixels inside a circular sector over the area of the circular sector.

Interior stroke distributions: With this feature extraction method, stroke distributions are calculated inside the outer contour (envelope) of an object. This kind of feature is utilized in the context of offline signature verification by Ferrer et al. [29]. Interior stroke distributions are calculated both in polar and Cartesian coordinates as features.

Chain code: Chain code represents a contour with coordinate of an arbitrary starting point and directions of transitions to reach the following points in the

contour until the starting point is visited again. If this procedure is realized for each blob in an image, then chain code can be used as a lossless compression algorithm. This encoding method is especially effective for images consisting of a reasonably small number of large connected components. Signature is a good example of it and this coding has been applied to offline signature verification as a discriminative tool [34, 35].

Modified direction feature: MDF utilizes the location and direction of transitions from background to foreground pixels. The direction information is integrated with a technique for detecting transitions between background and foreground pixels in the character image. It has found a wide variety of applications especially in handwriting problems since it has been proposed. It has been utilized for offline signature verification by Nguyen et al. [30].

Contour features: Contour features exploit curvature, direction co-occurrence and run-length information of contours. They have many applications such as image moments, contour area, contour perimeter, contour approximation, convex hull, minimum enclosing circle. They have been adapted to offline signature verification [36].

Projection features: Projection features are integrals of image in some direction (generally vertical or horizontal). Integral images are a fast way to compute the sum of a rectangular region of an image. The main advantage is that once the integral image is computed, sum of any rectangular region can be evaluated in constant time. They are used in offline signature verification [54].

High pressure points: High pressure points (HPP) are signature pixels which have gray level values upper than a threshold. Statistical distribution of HPP in polar coordinates is adapted to the problem by Vargas et al. [40]. Features representing information about pressure distribution from a static image of a handwritten signature are analyzed for an offline signature verification system by Vargas et al. [41].

Statistical texture features: Statistical texture analysis involves the computation of texture features from the statistical distribution of observed combinations of intensities at specified positions relative to each other in an image. Most common example is the GLCM utilized in conjunction with LBP. They are used successfully in offline signature verification [3, 42, 47].

Histogram of oriented gradients: Histogram of oriented gradients (HOG) is proposed by Dalal and Triggs [82]. It involves first computing the gradient information at each pixel inside a particular grid zone (either Cartesian or Polar). Next, histogram of gradient orientations in that zone is computed. We can conclude that HOG features utilize a coarse shape of signature by modeling local directions of gradients with histograms. HOG features are used by Zhang for offline signature verification problem [43].

Local binary patterns: Local binary pattern (LBP) is a powerful feature proposed to capture the texture in objects [83]. In the basic LBP method, a gray scale image is processed such that a binary code is generated for each pixel in the image. This code encodes whether the intensities of the neighboring pixels are greater or less than the current pixel's intensity. So, for instance in a 3x3 neighborhood with the current pixel being the center, a binary code of length 8 is generated consisting of 0s and 1s, according to the relative intensities of the neighbors. A histogram is then computed to count the number of occurrences of each binary code, describing the proportion of common textural patterns. LBP is very suitable for offline signature verification and has been utilized in several works [3, 48]. The reason is that, LBP encodes neighboring patterns of pixels well.

There are many LBP variants proposed in the literature. However, there are few works for LBP pattern selection proposed so far. An important drawback of the original LBP method is the sparse histogram generated, for example of size 256 for 3 by 3 neighborhood. Much of these patterns would never be seen on a small image sample. An example LBP histogram selection is applied to color texture classification by Porebski et al. [84]. It consists in assigning to each histogram a score which measures its efficiency to characterize the similarity of the textures within the different classes. The histograms are then ranked according to the proposed score and the most discriminant ones are selected. Selection is based on one of the simplest available methods according to the authors. It is a within-class histogram intersection similarity measure. Accuracy rates are reported to increase less than 0.5% in different color spaces.

There are plenty of works in literature to offer more compact histograms instead of pattern selection. In the work by Sujatha et al. [85], a special or operator is

implemented which takes or of symmetric neighbor pairs, claiming to preserve more than 90% of information content while reducing the LBP code to 4 bits. Another work to compactly represent exponentially growing circular neighborhoods is presented by Mäenpää and Pietikäinen [86]. Large-scale texture patterns are detected by combining exponentially growing circular neighborhoods with Gaussian low-pass filtering. Then, cellular automata are proposed as a way of compactly encoding arbitrarily large circular neighborhoods.

Because of the exponential growth of histograms, it is not feasible to directly encode farther neighborhoods with closer neighborhoods. A novel way to jointly encode multiple scales is proposed by Qi et al. [87]. When each scale is encoded into histograms individually, the correlation between different scales is ignored and a lot of discriminative information is lost. The joint encoding strategy can capture the correlation between different scales and hence depict richer local structures. Reported results show about 7% accuracy improvement over baseline multi-scale LBP on texture recognition problems.

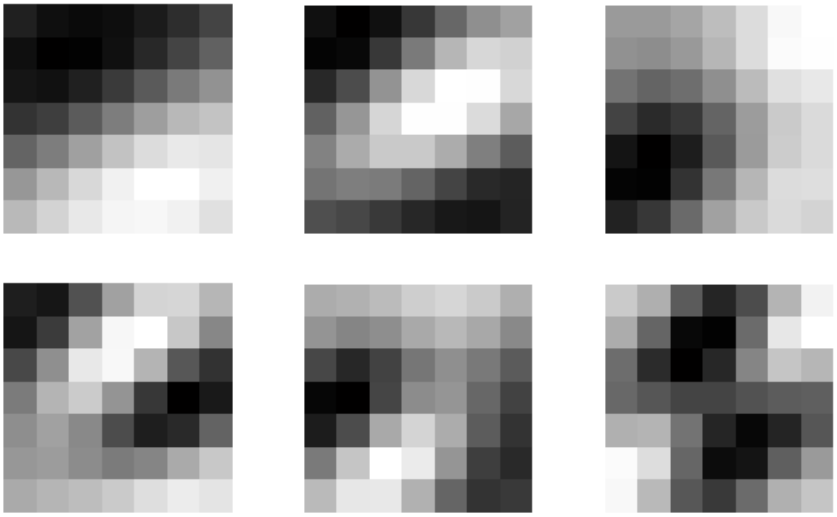
Zhang et al. offered a multi-block LBP method [88]. Inspired from Haar-like features [89], simple averaging in multiple rectangular blocks is applied to come up with 3 by 3 rectangular blocks of multiple pixels, each being treated like a single-pixel to calculate conventional LBP code. This method is capable of taking farther neighborhoods into account while avoiding the exponential growth in the resulting histogram. However, farther neighborhoods are taken into account in a coarse way of simple gray-level averaging. Performance improvement is expected to be low when working with binary images such as in the problem of offline signature verification.

Scale invariant feature transform: Scale Invariant Feature Transform (SIFT, [90]) is a popular feature extraction method used in computer vision. It finds distinctive, scale and rotation invariant features in images that can be used to perform matching between different views of an object or scene. It first extracts keypoints in images and then performs a matching between two images. SIFT features are used for offline signature verification [38, 91].

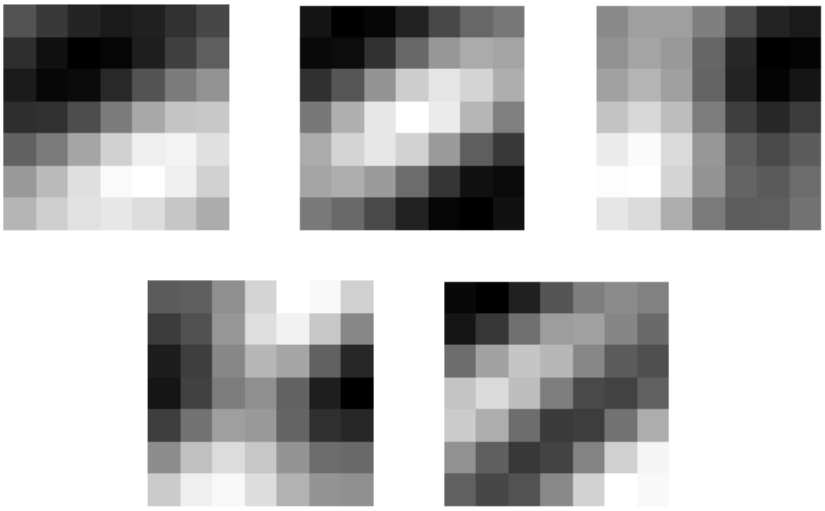
Speeded up robust features: Speeded up robust features (SURF, [92]) is a robust local feature detector that can be used in computer vision tasks like object recognition or 3D reconstruction. It is partly inspired by the SIFT descriptor.

The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT. SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images [93]. A novel signature stability analysis by using SURF is presented by Malik et al. [39].

Deep learning: Deep learning algorithms are proposed to learn the hierarchy of features in an unsupervised fashion, using large amounts of unlabelled data. We analyze the performance of a simple deep learning baseline for image classification, PCANet [94]. It comprises only the basic data processing components: cascaded principal component analysis (PCA), binary hashing and block-wise histograms. In the proposed architecture, PCA is employed to learn multistage filter banks. It is followed by simple binary hashing and block histograms for indexing and pooling. We obtain 20.37% EER with USVM (Section 4.2) of 5 references on GPDS-160, using the features learnt by PCANet. Best filters learnt are shown in Figure 3.1 for a 2-layer network. Deep learning algorithms promise to learn good features automatically from a given large data set, without manual work. However, there are still many parameters that are needed to be tuned with PCANet such as the parameters of the classifier (SVM as proposed by the authors), number of layers of the network, patch size, number of filters learnt in each layer, histogram block size, ratio of overlap for the blocks, fixed size of the signatures.



(a)



(b)

Figure 3.1: Filters learnt by a 2-layer PCANet, layer 1 (a) and layer 2 (b).

CNN is another way of learning good features automatically from training images. They are not studied much in the context of offline signature verification. An example work is proposed by Khalajzadeh et al. [50].

An example successful application of deep learning methods to biometrics is published by Sun et al. for face verification [95]. A hybrid convolutional network (ConvNet) - Restricted Boltzmann Machine (RBM) model for face verification in wild conditions is proposed. A key contribution is to directly learn relational visual features, which indicate identity similarities, from raw pixels of face pairs with a hybrid deep network. The deep ConvNets mimic the primary visual cortex to jointly

extract local relational visual features from two face images compared with the learned filter pairs. These relational features are further processed through multiple layers to extract high-level and global features. Multiple groups of ConvNets are constructed in order to achieve robustness and characterize face similarities from different aspects. The top-layer RBM performs inference from complementary high-level features extracted from different ConvNet groups with a two-level average pooling hierarchy. The entire hybrid deep network is jointly fine-tuned to optimize for the task of face verification. However, offline signature verification is a behavioral biometric and it differs from face verification task in the sense that in face verification different users constitute different classes but in offline signature verification different classes are constituted by genuine and skilled forgery samples of the same subject.

The reason such self-organizing feature learning methods are not successful in the context of signature verification may be found in the forgery definition of signatures. In other biometrics such as face recognition, there are no skilled forgeries except some fooling techniques such as showing face image of an enrolled subject to the camera. So in the context of non-behavioral biometrics such as face recognition, it is enough to find features that discriminate one subject's face from other subjects' faces. That kind of discrimination fits to the random forgery definition in the context of signature verification. We can think signature verification as a 2-class problem with genuine and forgery classes. Genuine and skilled forgery signatures of the same subject are very similar while these two are different classes, however genuine signatures and skilled forgery signatures of different subjects are completely different while being in same classes. One needs an extension for such algorithms to limit the learner within the signatures of the same subject while taking all subjects into account independently to come up with good global features.

Sparse dictionary: Sparsity-based approaches have proven to be very successful in many computer vision tasks in the last years. In image classification or detection, task-specific data is used to build a dictionary or codebook to represent images with sparse coefficients.

A recent investigation on the performance of sparse representation and dictionary learning for handwritten character recognition is done by Duong et al. [96]. In that work, sparsity-based approach was reported to under-perform the state of the art;

however, due to its performance over many other problems, we plan to investigate building a sparse dictionary of LBP codes for the signature verification problem, as part of future work.

Proposed features: We utilized a complementary set of features that are commonly reported successful in the context of offline signature verification, namely HOG, LBP and SIFT features. Our features are explained in detail, in Sections 3.3-3.5 after describing the grids used to extract local features in Section 3.2.

3.2 Grids in Cartesian and Polar Coordinates

Global features can be localized by dividing the image into regions and extracting the feature in such regions. Localizing the features exhibit global information when the features are combined, ending up with a more precise feature vector.

In order to develop a system robust to global shape variations, we extract features from local zones of the signature image. It is shown in most of the works that, localizing the features by the help of a grid superimposed on the aligned signatures yields to satisfying results. For this, the image is either divided into zones using a fixed number of rectangular grids in Cartesian coordinates or using a circular tessellation around the origin point in logarithmic-polar coordinates. This type of localization is utilized for offline signature verification previously, for example by Ferrer et al. [29].

Cartesian grids: First and most common choice is the use of rectangular grids in Cartesian coordinates. The grids may be overlapping to capture the signature at grid boundaries, or non-overlapping. A sample signature, overlaid with $10 \times 20 = 200$ non-overlapping rectangular grids is shown in Figure 3.2. A sample signature with 20% overlapping $6 \times 6 = 36$ grids is shown in Figure 3.3, grids are shown altogether. We use overlapping grids which are found to perform better.

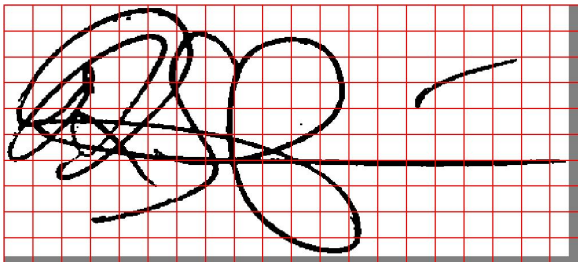


Figure 3.2: Cartesian non-overlapping grids.

Figure 3.3: Cartesian 6x6 20% overlapping grids shown altogether.

Log-polar grids: Another choice of coordinate system is the log-polar coordinate system. If the registration point is selected as the top-left point of the bounding box and the embellishments are on the right, then the left parts of the two signatures align better than the right. With this observation and at the cost of having some redundant features, we decide to use multiple registration points (center, top-left, top-right and so on) in the polar grid, to reduce the effect of registration mismatches. A sample signature divided into regions in log-polar space is shown in Figure 3.4 where the origin is taken as the image center. Same signature with overlaid log-polar grids where the top-left corner is used as the origin is shown in Figure 3.5.

The motivation behind using multiple fixed origin points in the polar coordinate system is that, there are no reference points in signatures, unlike face (eyes, nose tip etc.) or to some degree fingerprints (core point). The centroid or center of mass can be used as a lesser alternative in registering two signatures. Unfortunately, the location of both of these points may show large variations due especially to large variations in embellishment.

We select the origin points of log-polar coordinate system using a uniform pattern, to be independent of inter-user signature variations. We use a uniform point distribution pattern that has two parameters $count_x$ and $count_y$. We take $count_x$ number of uniformly spaced points horizontally; then we uniformly repeat this points

$count_y$ times, vertically. An example point distribution pattern where $count_x = count_y = 4$ is shown in Figure 3.6.

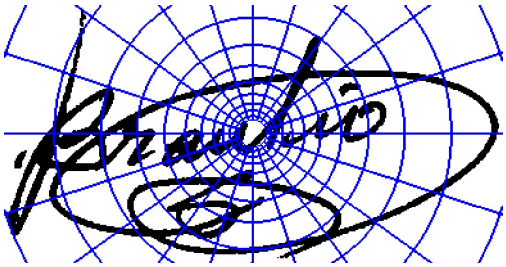


Figure 3.4: Log-polar grids, origin taken as the image center.

Figure 3.5: Log-polar grids, origin taken as the top-left corner.

Figure 3.6: Origin points selection pattern for log-polar coordinates.

Feature vectors: Once the grids (in rectangular or polar coordinate system) are fixed, the feature vectors are obtained by the concatenation of features extracted from each zone. Using a fixed grid addresses the problem of uniform scaling, however embellishments such as those at the beginning or end of a signature may significantly vary in location, orientation and size; thereby significantly changing the global shape of a signature and consequently its alignment to a reference signature.

Hierarchical representation: Using a small number of grids will end up in a global-like behavior of feature extraction and localization capability will be lost. In contrast, using too many grids will decrease the ability to allow for deformations. To eliminate the need for searching the ideal grid resolution, we use a hierarchy

of grids in increasing resolution, in order to extract coarse to fine features. In the top-level, the one grid corresponds to the full image, while in lower levels, higher numbers of grids are used.

Features extracted from all levels are concatenated at the end to form the final feature vector. This corresponds to concatenating coarse to fine number of distance bins, angular bins and origin points triples in polar space and concatenating coarse to fine number of grids in Cartesian space.

3.3 Histogram of Oriented Gradients

We use histogram of oriented gradients introduced by Dalal and Triggs (HOG, [82]) relative to the dominant orientation. The HOG features look at the gradient orientation histograms in a zone. While computing the gradient orientation histogram, we apply a circular shift normalization to allow for rotational differences of the strokes within the grid zone. Specifically, after finding the gradient orientation at each point, we find the dominant gradient orientation and represent it at the first bin of the histogram. Without this normalization, a rotation of the strokes in a zone would correspond to a circular shift in the HOG histogram; lowering the match between the original and matched histograms.

Note that while complex features give more information, simpler features such as gradient orientation are more robust to normal variations found in a signature. This method results in smaller feature vectors and as a result it is computationally efficient, while giving performance results comparable to more complex features.

HOG features are extracted both in Cartesian and Polar coordinates, separately.

3.4 Local Binary Pattern

LBP features compute co-occurrence of pixel values in predetermined neighborhoods. LBP method is commonly used in object recognition with good success and we expected it also to be useful in offline signature verification. Furthermore, since LBP is a texture feature, we expected it to be complementary to the HOG features that are also used in this thesis.

An example work that combines HOG and LBP features successfully was designed to detect partially occluded humans in scenes [97], reaching the best human detection performance on the INRIA dataset. LBP features are used in signature verification [3] as well.

In this thesis, LBP features are extracted only in Cartesian coordinates. We utilized the LBP method by different approaches explained below.

3.4.1 LBP-0

Conventional LBP method encodes all 2^8 neighboring types for a 3x3 neighborhood with 8-neighbor application, then for each of the codes we count the number of occurrences to generate a histogram. We extracted LBP-0 features both globally and in coarse to fine number of Cartesian grids. Baseline global LBP extraction is to be used for LBP patterns selection as will be explained in detail.

3.4.2 LBP-1

LBP-0 results in a sparse feature vector. Also after hierarchical grids extraction, feature vector gets bigger although the considered neighborhood is just 8-neighbors or $3x3$ neighborhood. Moreover, most of the patterns are never ever seen in a single grid. We make the system faster and concurrently improve the performance by considering just 4-neighbors ({South, North, West, East}) and diagonal neighbors ({North-East, North-West, South-East, South-West}) resulting in a feature vector of size $2 \times (2^4)$. This circularly symmetric grouping is inspired from the work by Ojala et al. [98]. Performance improvement is because of the sparse feature vector obtained from traditional LBP method (8-neighbor LBP-0) also ending up with higher complexity in terms of memory requirement and computation time.

We implicitly select 32 LBP patterns, however this kind of usage implicitly combines histogram of don't-care patterns (e.g. all combinations of diagonal-neighbors for each of 4-neighbor type). Example is provided in Figure 3.7 depicting the illustration of 4-neighbors where gray pixels are not cared so all possible 16 combinations of gray pixels (diagonal-neighbors) are combined into the histogram of each pattern obtained from black pixels.

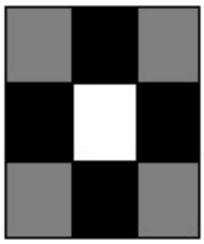
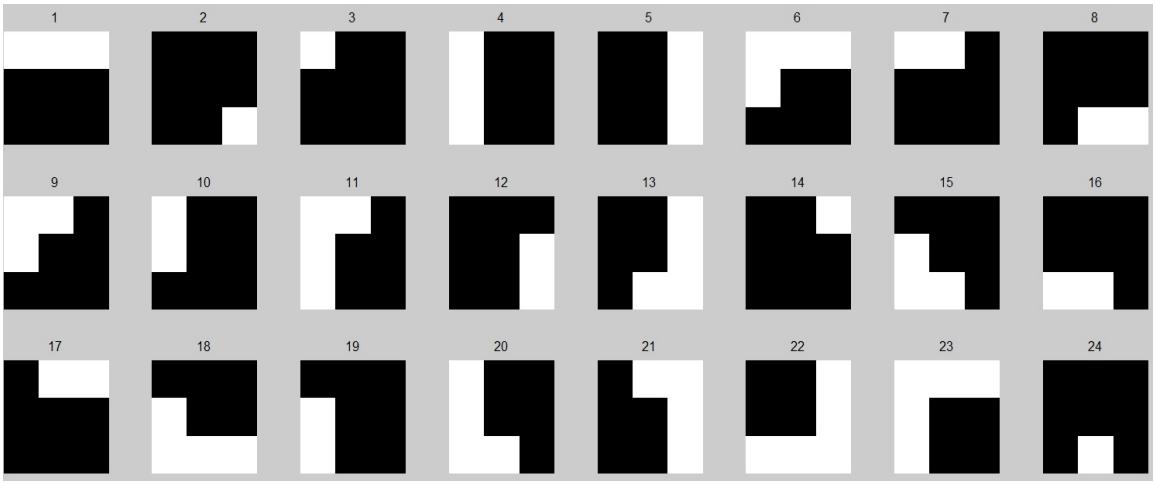


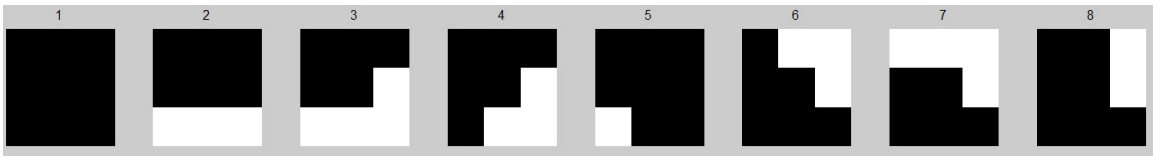
Figure 3.7: Each 4-neighbor implicitly combines all combinations of diagonal neighbors.

3.4.3 LBP-2

In this method we take all patterns like in LBP-0, but select the best patterns explicitly. Selection criterion is based on term frequency of each pattern. In a training set (GPDS 161-300), we collect mean of all genuine samples' LBP-0 feature vectors as Gen_{global} and mean of all skilled forgery samples' LBP-0 feature vectors as $Forg_{global}$. Then we compute $|\Delta TF|$ for each feature where $\Delta TF = Gen_{global} - Forg_{global}$. We select first 32 features with highest $|\Delta TF|$ value. Selected 32 patterns that are found more frequently in genuines and that are found more frequently in forgeries are shown in Figure 3.8.



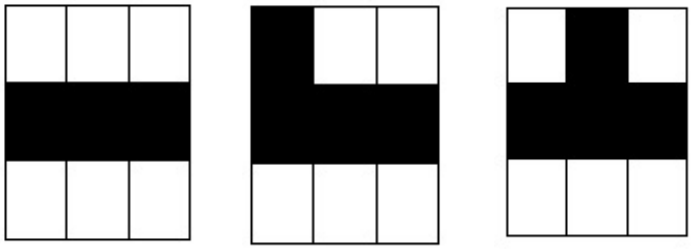
(a)



(b)

Figure 3.8: 3x3 patterns with highest ΔTF values (a) Positive ΔTF (more frequent in genuines) (b) Negative ΔTF (more frequent in forgeries). Black pixels represent on (pencil) pixels.

Computing the histogram: Although with this kind of usage we actually select 32 patterns; when computing a particular histogram entry H, we combine the count of all patterns that are 1-pixel away from H to obtain a dense histogram like the dense histogram of LBP-1. Histogram bins of helping patterns are combined using a small weight (0.2) to obtain the final dense histogram entry H. An example selected pattern and 2 helping patterns with 1-pixel distance are shown in Figure 3.9.



(a) (b) (c)

Figure 3.9: Histogram generation (a) Example selected pattern (b) A helping pattern (c) Another helping pattern.

Alternative pattern selections: To explore the effect of selecting the best patterns according to term frequency criterion; we also generate LBP2$_{min}$ features by selecting the worst 32 patterns with smallest $|\Delta TF|$ values, and LBP2$_{rnd}$ features by selecting random 32 patterns. Random pattern selection is repeated several times. LBP2$_{rnd\&averageEER}$ denotes the average EER obtained from random pattern selection experiments. LBP2$_{rnd\&scorefusion}$ denotes simple averaging score level fusion over various random pattern selection experiments. Not the first 32 patterns with highest $|\Delta TF|$ value, but the next 32 patterns are also selected for experimentation. This is denoted as LBP2$_{n32}$.

3.4.4 LBP-0F

It is possible to generalize the LBP idea to bigger neighborhoods than 3x3. Detecting LBP patterns on a larger window can be useful, but in that case the number of patterns grow significantly. For 5x5 window, there are 2^{24} patterns. For that reason, we just consider the borderline pixels. When we consider a farther neighborhood such as Chebyshev distance 2 corresponding to 5x5 window, we consider all the patterns constructed just by 2-Chebyshev distance pixels as shown in Figure 3.10, ignoring the variations in the 3x3 center.

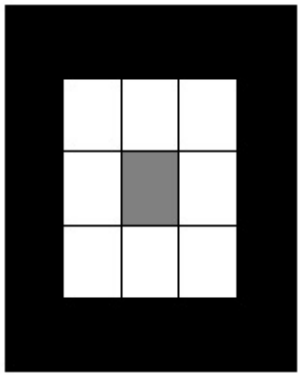
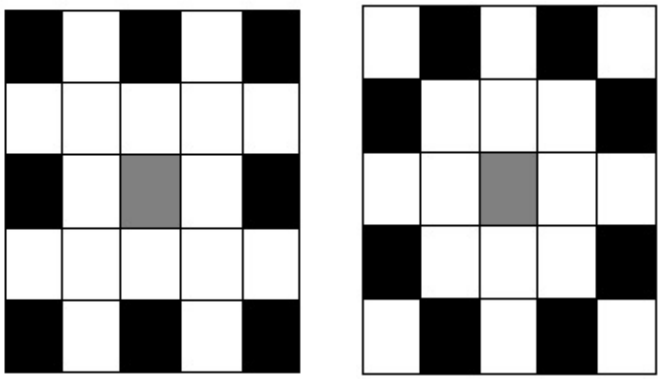


Figure 3.10: Neighbors with Chebyshev distance 2 in black, center pixel shown in gray.

Reducing the feature size: This results in 2^{16} patterns, which is difficult to deal with in practice. The generalized LBP operator is derived on the basis of a circularly symmetric neighbor set of a defined number of members on a circle of radius R [98]. This LBP operator is applied to offline signature verification by

Vargas et al. [3]. In order to reduce the number (2^{16}), we sample the pixels of 2-Chebyshev distance resulting in several groups of 8 pixels (2 groups for 2-Chebyshev distance). Example pixel groups are illustrated in Figure 3.11 for 5x5.



(a) (b)

Figure 3.11: LBP-0F neighbors with Chebyshev distance 2 sampled in 2 groups, each group having 8 pixels.

Combining the groups: We build 2 separate classifiers (USVMs - Section 4.2) for each sample group. We then generalize the idea to 7x7 where we have 24 pixels of Chebyshev distance 3, grouped into 3 groups. These features are to be combined in score level where a simple averaging is applied. This overall feature extraction and classification mechanism is named LBP-0F.

3.4.5 LBP-1F

We follow the idea in LBP-1 of grouping the pixels into groups such that each group has 4 equidistant pixels as opposed to LBP-0F where there are several groups each having 8 equidistant pixels. Example pixel groups are illustrated in Figure 3.12 for 5x5. There is no pattern selection in LBP-1, so each group has limited number of (4) pixels to prevent the feature vector from becoming too large. These 4 circularly symmetric groups are again inspired from the work by Ojala et al. [98], containing completely independent neighbors of Chebyshev distance 2 and covering all of the neighbors with Chebyshev distance 2, forming a basis.

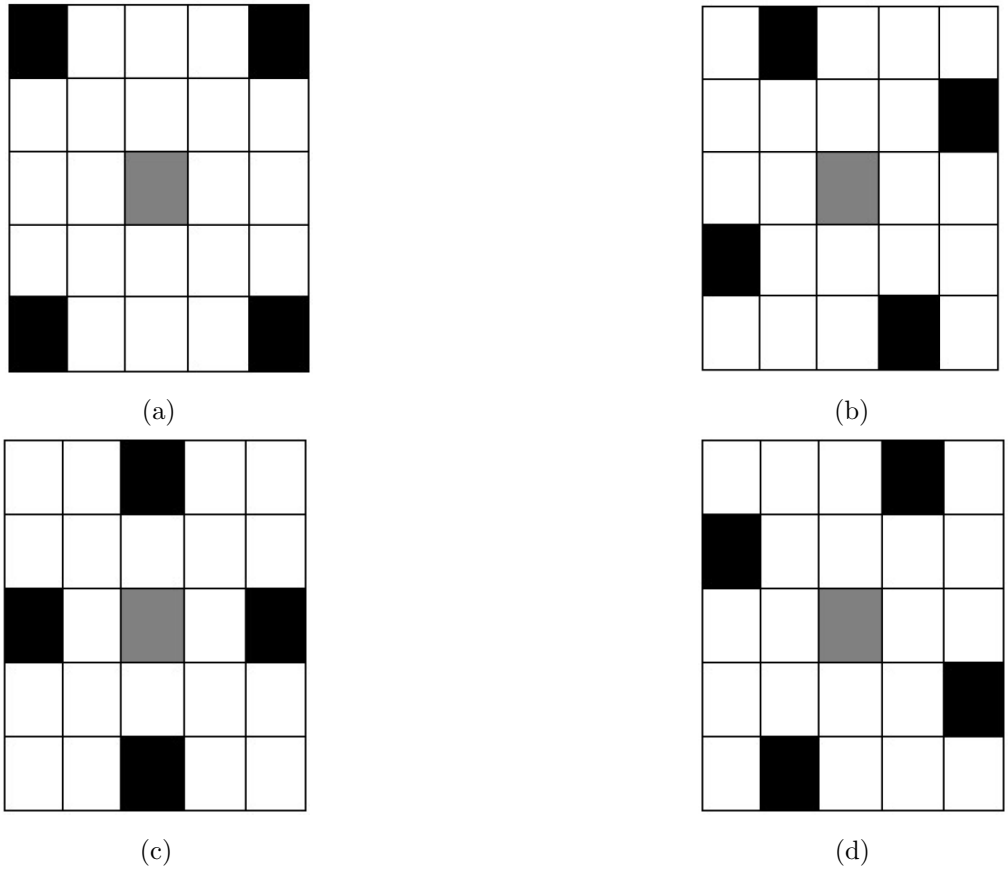


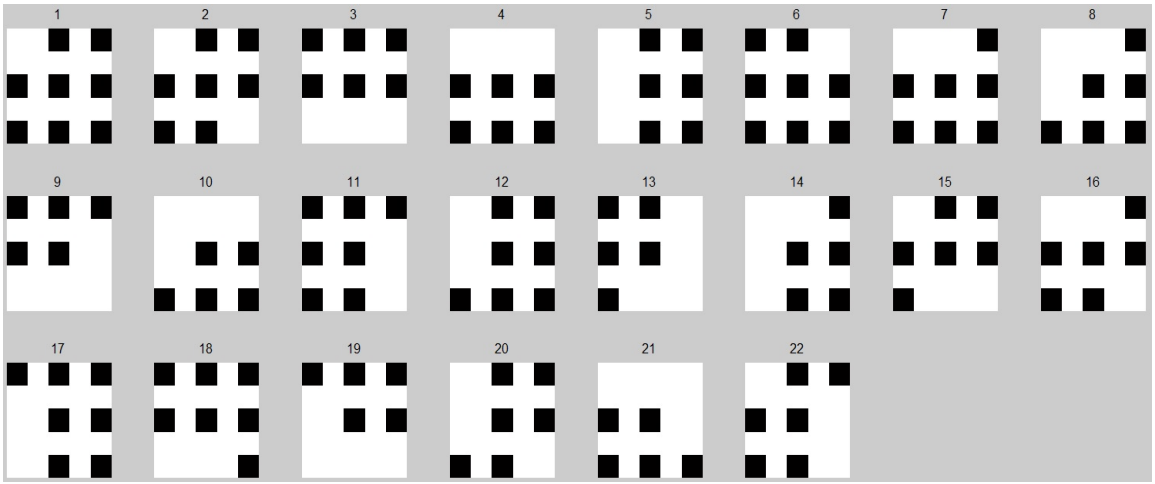
Figure 3.12: LBP-1F neighbors with Chebyshev distance 2 sampled in 4 groups, each group having 4 pixels.

Because there are 16 pixels of Chebyshev distance 2, there are 4 groups of 4 equidistant pixels, as opposed to that of 2 groups in basic LBP-1. We name the LBP feature that is extracted by grouping 16 pixels of Chebyshev distance 2 as $LBP-1F_{5x5}$. We generalize the idea to 7x7 where we have 24 pixels of Chebyshev distance 3 grouped into 6 groups with 4 equidistant pixels, named $LBP-1F_{7x7}$. We further generalize the idea and obtain $LBP-1F_{9x9}$. These features are to be combined in score level where a simple averaging is applied. This overall feature extraction and classification mechanism is named LBP-1F.

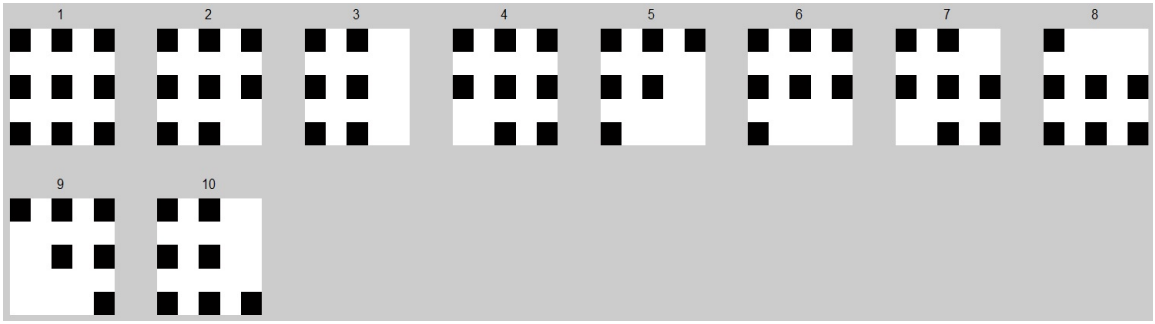
3.4.6 LBP-2F

Selection of good LBP patterns in LBP-2 can also be applied to farther neighborhoods. For example, pre-selected specific paths of Chebyshev distance 2 are utilized for the purpose of offline signature verification [99]. We select the best patterns for each group of each distance where the neighbor sampling is done as in LBP-0F.

Pattern selection is done as explained in Section 3.4.3. Best 32 patterns for the first sample are shown in Figure 3.13, whereas best 32 patterns for the second sample are shown in Figure 3.14.

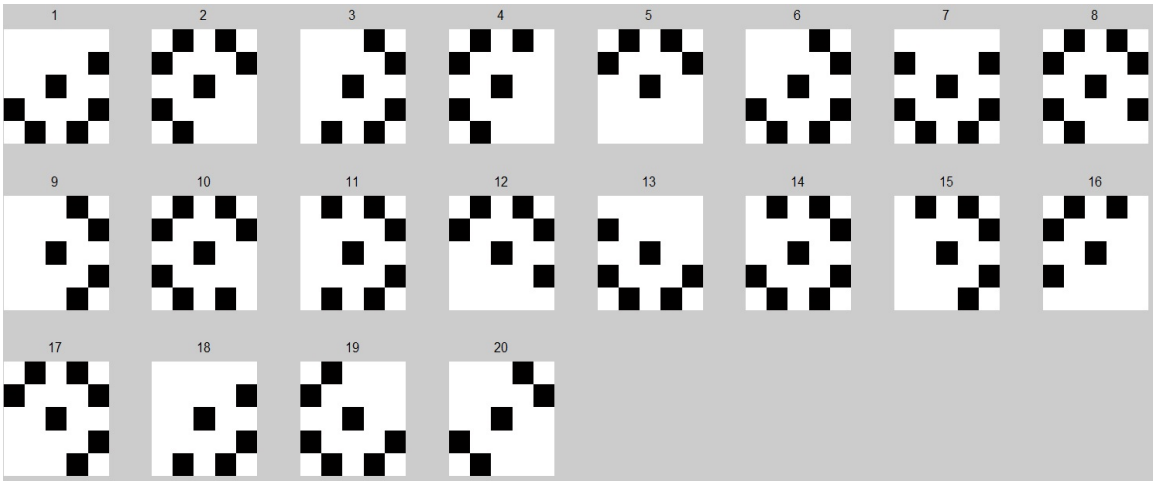


(a)

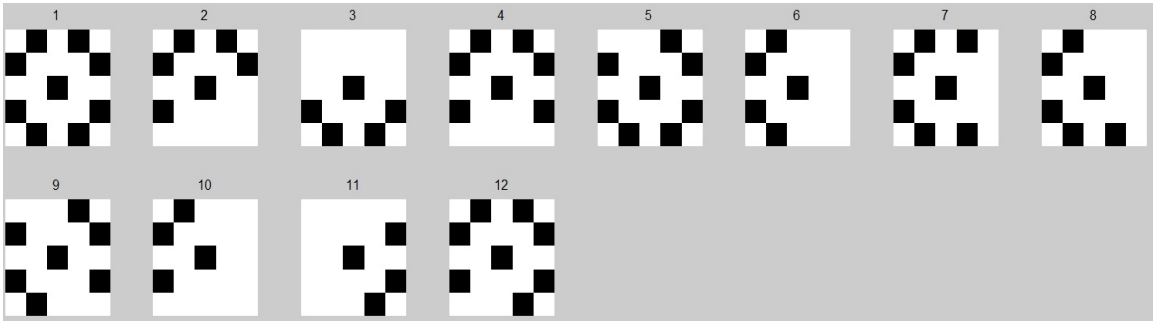


(b)

Figure 3.13: 5x5 sample 1 patterns with highest ΔTF values (a) Positive ΔTF (more frequent in genuines) (b) Negative ΔTF (more frequent in forgeries). Black pixels represent on (pencil) pixels.



(a)



(b)

Figure 3.14: 5x5 sample 2 patterns with highest ΔTF values (a) Positive ΔTF (more frequent in genuines) (b) Negative ΔTF (more frequent in forgeries). Black pixels represent on (pencil) pixels.

Combining the groups: We build separate classifiers (USVMs - Section 4.2) for each sample group. Count of 1-pixel away patterns are combined in the generated histogram like described in LBP-2. We further generalize the idea to 24 pixels of 3-Chebyshev distance sampled in 3 groups with equal-distance of 2 pixels. Having 2 independent classifiers for each sample group of each distance level, we totally have 6 classifiers each one being an expert on completely independent information. We use the average of 6 classifier scores to have a final score for LBP-2F.

Alternative pattern selections: To explore the effect of selecting the best patterns according to term frequency criterion; we also generated LBP2F_{min} features by selecting the 32 patterns with lowest $|\Delta TF|$ value, and LBP2F_{rnd} features by selecting random 32 patterns. Random pattern selection is repeated several times. LBP2$F_{rnd\&averageEER}$ denotes the average EER obtained from random pattern selec-

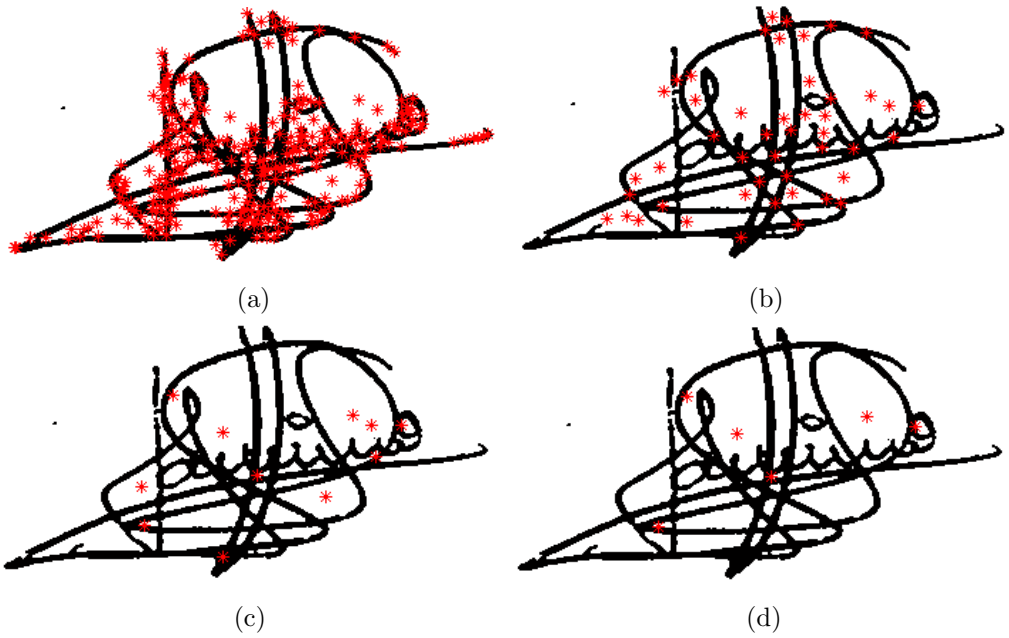


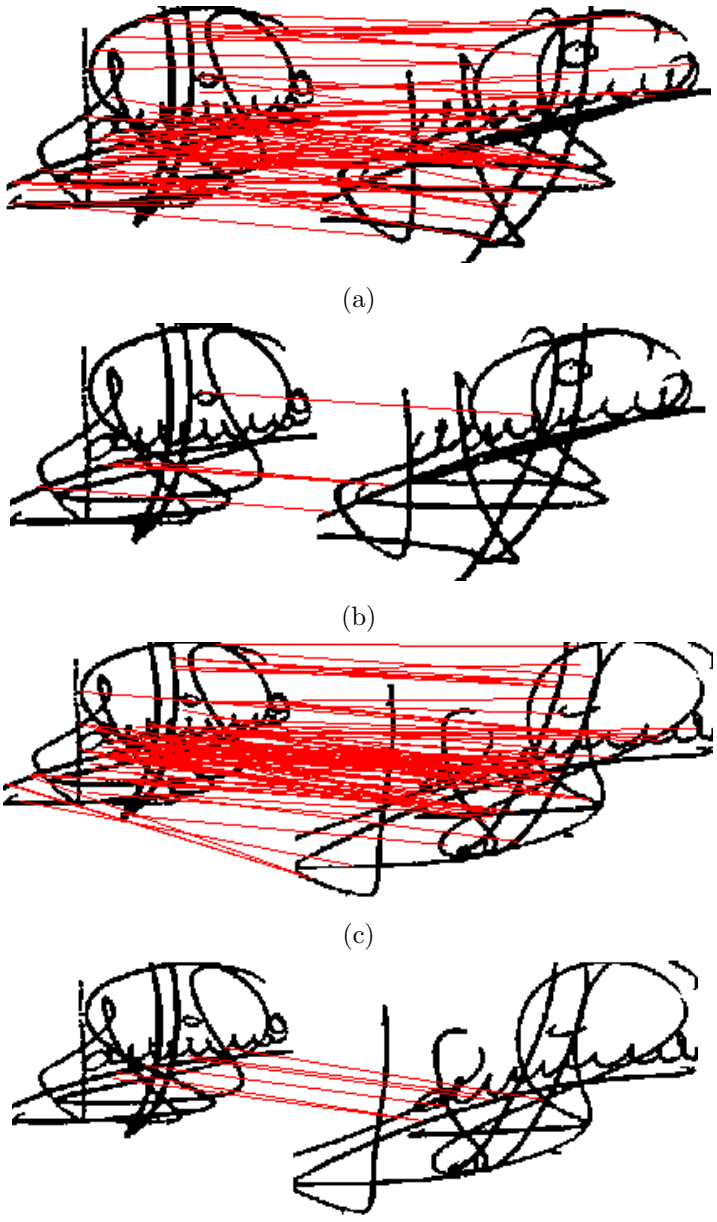
Figure 3.15: Example SIFT keypoints thresholded with respect to scales.

tion experiments. LBP2$F_{rnd\&scorefusion}$ denotes simple averaging score level fusion of individual LBP distance-sample classifiers over various random pattern selection experiments. Not the first 32 patterns with highest $|\Delta TF|$ value, but the next 32 patterns are also selected for experimentation. This is denoted as LBP2F_{n32}.

3.5 Scale Invariant Feature Transform

SIFT method comes with keypoint extraction and keypoint matching functionalities. Example SIFT keypoints extracted from a signature, thresholded with respect to scales of the keypoints are shown in Figure 3.15. In conventional SIFT keypoint matching, a common rigid transformation is found. We discretize the SIFT matchings as separate rigid transformations and analyze the performance of using the number of votes in the most populous transformation. Example matches between two signature pairs are shown in Figure 3.16; corresponding matches of the most populous transformations are separately shown.

The transformation parameters are found as follows. Suppose that I_1 and I_2 are two images to be matched, x_1, y_1, x_2, y_2 are corresponding coordinate vectors of matches provided by the SIFT algorithm. We first find the normalized coordinates $xn_1 = x_1/w_1, yn_1 = y_1/h_1, xn_2 = x_2/w_2, yn_2 = y_2/h_2$ where w_i and h_i are the



(a)

(b)

(c)

(d)

Figure 3.16: Example SIFT matches (a) and (c), corresponding orientation-translation matches of the most voted transformation (b) and (d).

width and height of image i. We find the translation in two dimensions using $xd = xn_1 - xn_2$ and $yd = yn_1 - yn_2$. We then find orientations of matches using $\theta = arctan((y_1 - y_2)/(x_1 - x_2))$. We quantize θ values into 8 bins, xd values into 4 bins, yd values into 4 bins; in total 128 bins.

There are several alternatives to normalize the number of matches in the highest voted transformation bin N_h to be able to use this number as the score for classification. We investigate two different normalization methods. We refer the first normalization method **SIFT-MP**, where we simply divide N_h to total number of matches N and use N_h/N for classification. SIFT-MP corresponds to match counts normalized as percentage. We refer the second normalization method **SIFT-MR** where we find average N_h over all possible reference by reference matches as N_h^R and use this number for normalization: N_h/N_h^R. SIFT-MR corresponds to match counts normalized with reference counts.

One can easily observe that there are non-linear deformations between signature images. Especially with signatures having affluent embellishments, different parts of the signature may register differently. To handle such situations, we utilize a novel method which we refer **SIFT-TH**. In SIFT-TH feature extraction, we use the number of votes in all transformation bins to generate a histogram which combines orientation and translation bins of matches between two images, treated as a feature vector. When we combine the transformation bins into a single histogram, we get a feature vector of size $8 \times 4 \times 4 = 128$. We use the number of match points in each transformation bin as a feature vector; in other words, we have a 128-dimensional feature vector containing the number of matches in each transformation. This novel representation is intended to address signatures where two parts of a signature may undergo different transformations. For instance for the genuine signatures of a person who signs his signature without any variability in the main body but a lot of variation in the embellishing stroke, the transformation histogram will show a consistent high match in one bin (0 rotation and 0 translation) and a smaller match in one of the other bins.

We obtain both the normalization number N_h^R of SIFT-MR and training SIFT match histograms of SIFT-TH described above by applying reference by reference matches as follows: Suppose N is the number of references, R_i is the reference

with index i. We collect leave-one-reference out inter-reference matches to use as positive examples, such as (R_1, R_2), (R_1, R_3), (R_2, R_1), (R_2, R_3), (R_3, R_1), (R_3, R_2) for $N = 3$. We intentionally use matches both ways so that the classifier can learn that features obtained from both type of orientations / translations are genuine.

In addition to reference by reference (genuine) training matches, we need histogram samples obtained by matching random forgeries to references with SIFT-TH where we use USVM verification protocol (Section 4.2). Suppose Q is a random forgery (forgery of another user). We find two-way matches between all references and Q (such as (R_1, Q), (R_2, Q), (R_3, Q), (Q, R_1), (Q, R_2), (Q, R_3)...). We then collect all of such matches as negative examples.

While testing any of the three methods described above we get scores for all matches (R_1, Q), (R_2, Q), (R_3, Q), (Q, R_1), (Q, R_2), (Q, R_3) and use the median score as the final SIFT score for Q. In test case, Q can either be a genuine forgery or a skilled forgery of the user being tested.

We analyze the performance of three methods and also the effect of having finer transformation bins with SIFT-MP. We use 5 genuine signatures as the reference set on GPDS-160. Results are shown in Table 3.1.

Method	θ bins	# of x-bins and y-bins	EER
SIFT-MP	8	4	29.12%
SIFT-MP	16	6	33.60%
SIFT-MR	8	4	25.84%
SIFT-TH	8	4	24.09%

Table 3.1: SIFT results with different usages.

Chapter 4

Classification

One can use both user-based and global classifiers in offline signature verification. Because user-based classifiers are purposed to discriminate just a single person, they are reported to be more successful [100] with the requirement of enough references from each subject.

There are many different classifiers that are used in offline signature verification. For basic feature types and relatively easier problems, normalized Euclidean distance between features might be enough to do classification [29]. We investigate the performance of Euclidean distance; Euclidean distance with user-based normalization such as normalization by dividing to weighted median between references, dividing to mean distance between references; reference variance normalization and calculating multiple metrics such as mean of norms of distances to references divided by mean of leave-one-out inter-reference norms, maximum of norms of distances to references divided by maximum of leave-one-out inter-reference norms. Bayes classifier is used in a few works [38, 48]. K-nearest neighbor (KNN) classifier is one of the simplest choices and used for offline signature verification [26, 40]. HMM is heavily used in handwriting recognition and it is also popular in offline signature verification context [31,32,34,58,59]. Another possibility is to use neural networks [30,40,48,57]. We also investigate the performance of basic neural networks. However, SVM classifier [101] outperforms all other classification methods which is found very successful in signature verification [29, 30, 35, 47, 48, 58, 61]. In our system, classification is performed using SVMs, where two different approaches to train the classifier are investigated, namely global and user-dependent SVMs.

User-dependent SVMs (USVMs) are separately trained for each subject to learn

to differentiate that user's signature features from others'. In contrast, global SVMs (GSVMs) are user-independent classifiers trained with differences observed between query and reference feature vectors, across all training users. Simply speaking, they are meant to model deviations observed in forgery signatures.

Both classifiers are trained with RBF kernels and parameters are optimized with grid search on a separate validation set (users 161-300 from the GPDS-300 dataset, who are not in the test set). The number of genuine signatures used as reference is kept variable (5 or 12). For global SVMs (GSVMs), half of the users in validation set is used for training and the other half is used for testing. Linear SVM is also taken into account. It is experimentally found from USVMs that RBF kernel outperforms linear SVM for every feature type; 1.5% better for HOG grid, 3% better for polar grid, 12% better for LBP-2, 4.5% better for SIFT.

Combining user-dependent and global verification systems have been investigated before [102]. However, that approach focus on fusion of different biometric modalities in local and global domains, instead of local and global types of classifiers. Another example is by Eskander et al. [63], where a hybrid writer-independent (WI) and writer-dependent (WD) offline signature verification system is proposed. But WI and WD classifiers are selectively used in that work, instead of a concurrent usage. We follow the approach described in our previous work [100].

4.1 Global SVMs (GSVM)

A global (also called writer-independent or user-independent) signature verification system learns to differentiate 2 types of classes: genuine and forgery. A global offline signature verification system is proposed by Santos et al. [62]. A hybrid writer-independent (WI) and writer-dependent (WD) offline signature verification system is proposed by Eskander et al. [63]. One of the WI or WD classifiers is selectively used, depending on the number of references provided by a user. Hu and Chen [47] also separately use global and writer-dependent classifiers. We concurrently apply user-dependent and user-independent classification in our system. In the first approach, we train a global SVM which is a user-independent classifier trained to learn to separate difference vectors obtained from genuine signatures of a user, from those

obtained from (skilled) forgery signatures of the same user.

To obtain the difference vectors, features obtained from a query signature (genuine or forgery) are compared to the features obtained from each of the reference signatures of the claimed identity. The resulting difference vectors are then normalized so that each element of this vector represents how many standard deviation away the query feature is from the reference feature.

For the global classifier (GSVM), a standard deviation normalization scheme is applied, as stated in our previous work [100]. However in that work, normalization is done using standard deviation among the references. In this thesis, we improve the stated approach by applying the normalization using standard deviation among the difference vectors of a given query and references as follows:

More precisely, let $\{R^1, R^2, ..., R^N\}$ be the feature vectors extracted from the reference signatures of a particular user and let $Q = [q_1 \dots q_M]$ be the feature vector extracted from a test signature, where N is the number of reference signatures and M is the number of features. Then, we compute N difference vectors for each query, where the i^{th} difference vector is computed as:

$$D^i = Q - R^i = \begin{bmatrix} (q_1 - R^i_1)/(\sigma_1 + \tau) \\ (q_2 - R^i_2)/(\sigma_2 + \tau) \\ ... \\ (q_M - R^i_M)/(\sigma_M + \tau) \end{bmatrix} \tag{4.1}$$

where σ_m is the standard deviation of $q_m - R^i_m$ among the m^{th} feature of the difference vectors between query and claimed user's reference signatures $i = 1...N$, explicitly written as:

$$\sigma_m = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(q_m - R^i_m - \mu_m)^2} \tag{4.2}$$

where $\mu_m = \frac{1}{N}\sum_{i=1}^{N}(q_m - R^i_m)$, τ is a small constant to eliminate division by zero to handle the case where a specific feature difference remains the same among all differences. We can conclude that σ_m is thus calculated query-specific. By the help of this normalization, the difference vector represents how many standard deviations away the query feature is from the reference feature.

Because we have N difference vectors $Q - R^i$ obtained from each reference, we have N classifier scores for each query. Let $S(D^i)$ be the GSVM classifier score for the difference vector D^i which is calculated via reference R^i. To get a final classifier score, we calculate the average score value $(S(D^1) + ... + S(D^N))/N$.

We devote some of the users who are not in the test set (users 161-300 from the GPDS-300 dataset), and use all of their signatures (genuine and skilled forgery) to train the system. It is important to underline that, in this way, no skilled forgeries belonging to users in the test set, are used during training.

Note here that the SVM is learning which changes in the feature vector may be within the normal variations of a signer and which changes indicate forgeries. This can be better explained considering the case of a system using global features where the SVM learns how much variation in a particular feature (e.g. size, pixel density, width-to-height ratio) matters. In the case of local features, the SVM can learn how to weight differences in the center versus periphery of the signature for instance. While it is less intuitive in the case of local features, it is meaningful with user-dependent normalization and we have found experimentally that the combination of GSVM results improves accuracy.

Because the reference signatures are just used as pivots to generate the necessary difference vectors, few number of references is actually enough to test the GSVM. It is possible to evaluate a given query even with one reference of the claimed user. This is especially an advantage for real life cases.

It is important to emphasize that in the GSVM approach we do not build a user specific system but a general system to discriminate any user's genuine signature from that user's skilled forgery signature. Doing that, we do use skilled forgeries in the training phase; but we do not use any test user's any signature, instead we use signatures of (prior) training users (161-300) that we already have before enrolling any test user to the system. That is quite natural as any biometric system developer can buy or collect his/her own private database and build a general model prior to the release of that system. Actually, by the condition of doing appropriate image preprocessing, training users can be selected from a completely different database. In test phase, references of test users are just used to calculate difference vectors. All other signatures of test users are considered as test queries.

We follow the same train and test protocol and devote users 161-300 from the GPDS-300 dataset for GSVM training, and use all of their signatures to train the system so that no signatures belonging to users in the test set are used during training. Actually the users devoted for training could be selected from a completely different database if appropriate image normalization is applied. In testing, references of test users are just used to calculate difference vectors. All other signatures of test users are considered as queries to test the system.

GSVM learns which deviations in features are caused by the signer and which deviations are caused by forgers. In the case of global features, GSVM learns how important a deviation is in a single feature. In the case of local features, GSVM would learn how to weight deviations in different locations (like corners or center) of the signature.

Compared to USVM, GSVM comes with a disadvantage of the need for storing user signatures along with the GSVM model to be able to calculate the difference vectors. A method for adapting user-independent systems to different users is proposed by Eskander et al. [103], leading to secure and compact user-dependent systems. Feature representations embedded within user-independent classifiers are extracted and tuned to each enrolled user while building a user-specific classifier, in the stated work. However, GSVM proposed in this thesis does not need any further step for adaptation of enrolled users. We also get rid of the problem of varying user-based scores as much as possible with the help of difference vector standard deviation normalization.

4.2 User-dependent SVMs (USVM)

In the second approach, we train user-dependent SVMs, one for each user, with the expectation that the user-dependent SVM can learn to differentiate genuine signatures of a person from forgeries. For this, each SVM is trained with the raw feature vectors obtained from the reference signatures of the corresponding user and those obtained by random forgeries (other users' reference signatures reserved for training). Note that in this case, we do not need a separate group of users for training as opposed to GSVM, since we only use genuine signatures of others.

Using other users' genuine signatures as random forgeries during training can be avoided by the help of one-class classifiers. This corresponds to expert examiner of signatures. Expert examiner performs the verification by comparing the questioned signature to the references and then gives the decision according to the comparison. Another example of natural one-class classification is the process of object recognition such as recognizing an apple without comparing it to other fruits. An example work is proposed by Murshed et al. [104] for offline signature verification. A priori knowledge of class of forgeries is avoided by the help of cognitive information learning of fuzzy ARTMAP neural network.

4.3 User-based Score Normalization

Because it is obligatory to use a single threshold to do all users' verification decisions in a real-life scenario by using the definition of EER, it is useful to normalize user scores aiming to bring them to similar levels. Nearly all of the works in biometrics literature do score normalization by directly normalizing the scores with a Gaussian assumption with the help of genuine and forgery scores for each user. For a biometric like offline signature where skilled forgeries are possible, it is necessary to use many skilled forgeries and many references (genuines) to do such a normalization (like z-norm or t-norm) effectively.

In the work by Fierrez-Aguilar [105], a framework for user-dependent score normalization collecting previous work in related areas is provided and applied to online signature verification. They classify the normalization techniques into impostor-centric, target-centric, and target-impostor techniques. In impostor-centric methods (IC) no information about client (genuine) score intra-variability is used. In Target-Centric methods (TC) no information about impostor (forgery) score variability is used. In target-impostor methods (TI) information from both client score intra-variability and impostor score variability is used. For example, a TI normalization can be done using the following formula:

$$S_{TI} = s - (\mu_I \sigma_C + \mu_C \sigma_I)/(\sigma_I + \sigma_C) \tag{4.3}$$

where μ_C and σ_C can be obtained from the collection of classifier scores obtained with

a leave-one-reference out training and testing with the left-out references, whereas μ_I and σ_I can be obtained from skilled forgeries (real impostors) reserved for training or from random forgeries (casual impostors) obtained from other users. We do not use any skilled forgeries in training stage as one of our aims is to develop a real-life compatible system. We apply the normalization methods described in that work to our system, only with casual impostor and client score statistics. However we could not succeed to improve our results, which coincides with the conclusion of the authors: using casual impostor statistics for estimating the normalization functions leads to the highest performance improvement when testing with random forgeries but lowers verification performance in case of testing against skilled forgeries. If using real impostor statistics is an option, it leads to verification performance improvements when testing either with random or skilled forgeries.

Another user-based score normalization scheme is applied to offline signature verification by Panton [32]. Because it is not possible to estimate the score distribution of negative signatures for each enrolled client (writer), a Z_p score normalization is implemented. A prior knowledge that the score distributions are approximately Gaussian is required which is confirmed by the author. Z_p score normalization basically uses genuine reference score statistics (mean and standard deviation) to normalize the scores of queries.

We propose a novel score normalization method which is suitable for offline signature verification. We build a 3-class classifier which learns the direction of the shift: shift positive, no shift and shift negative. Positive and negative shift amounts are very small fixed numbers. They can also be optimized with the help of a validation set. We call this classifier SSVM (shift SVM). Training is done in a completely different set. EER thresholds for training and test sets might be different, say t_{train} and t_{test}. However, this should not affect the result estimated by the SSVM. We can trivially make the EER threshold of test set equal to t_{train} by adding $t_{train} - t_{test}$ to each of test scores and then apply the shift estimated by the SSVM. Naturally, we can apply the estimated shift to test scores directly. It is also possible to use a single SSVM to be used after score fusion, or individual SSVMs to be used for each classifier's scores before the score fusion stage. We empirically find that using individual SSVMs performs better.

We employ two types of features to utilize for SSVM: Basic measures from reference set and scores obtained from cohort set by testing the cohorts with USVMs trained with reference set.

Measures from the reference set includes variance of leave-one-reference-out feature distances. Let R_i, $i = 1, 2, ..., N$ be the features extracted from the reference set where N is the number of references. $\{R_i - R_k\}$ is the set of difference vectors where i is the left out reference and $k \neq i$. We use the variance of $max(\{R_i - R_k\}) - min(\{R_i - R_k\})$ among all left-out references i. Other measures of reference set are convex hull pixel density statistics of reference set, width and height statistics of references and leave-one-out USVM reference scores which are obtained as follows: Separate small USVMs are trained without a selected reference, that is $\{R - R_i\}$. S_i, the score for R_i is obtained from that USVM. All scores obtained this way are collected as $\{S_1, S_2, ..., S_N\}$ and the mean is calculated as S_{mean}.

Other information used as feature is the USVM scores obtained with the help of a distinct cohort set, which constitutes the most of the feature vector. For a user c in cohort set, we test the references with USVM of user i and use the statistics of the scores. Then we build another USVM for user c and test the references of i with that cohort's USVM. We also use the statistics of scores obtained that way.

While testing the SSVM, we get 3 probability estimates from SSVM for 3 shift classes (shift positive, no shift and shift negative). However, probability estimates are not distinctive enough to be used directly. We train a basic tree and prune it with the SSVM probabilities of training set to fit a rule to estimate the real shift class. We give more priority to no shift class as making no shift will be better than making a shift in wrong direction.

Full system with 12 references tested on GPDS-160 improves the EER 1.24% when all user scores are shifted ideally (with shifts to match their ideal thresholds). With the above mentioned shift estimation method, we currently have 0.3% EER improvement. However this method is not applied in our final system as the complexity is high regarding the small amount of improvement.

4.4 Classifier Combination

In most of cases, combining the classifiers of several sources is reported to perform better than feature level combination or feature selection methods while depending less on manual engineering work at system design level. In general, classifiers may differ by changing the training set, input features and parameters of the classifier. For example in an early work by Sabourin and Genest, design of the integrated classifiers is based on a large number of individual classifiers in an attempt to overcome the need for feature selection. Systematical evaluation of a multi-classifier-based approach for off-line signature verification is presented. Two types of integrated classifiers based on kNN or minimum distance classifiers and 15 types of representations related to the extended-shadow-code (ESC) used as a shape factor have been evaluated [106].

Receiver Operating Characteristic (ROC) curves are used for classifier combination to improve the performance by Oliviera et al. [107]. The contribution of the paper is two-fold. Different fusion strategies to combine the partial decisions yielded by the SVM classifiers are analyzed. Then ROC produced by different classifiers are combined using maximum likelihood analysis, producing an ROC combined classifier. Authors demonstrate that ROC combined classifier based on the writer-independent approach reduces FRR while keeping FAR at acceptable levels.

A multiple classifier combination applied to offline signature verification is proposed by Batista et al. [108]. In the first stage, a set of discrete HMMs trained with different number of states is used to calculate similarity measures that generate new feature vectors. In the second stage, these vectors are employed to train a SVM (or an ensemble of SVMs) that provides the final classification. Proposed system reduces the overall error rates when compared to a traditional feature-based system using HMMs. Later the same authors utilize an EoC for offline signature verification [109]. Two dynamic selection strategies are proposed, using the classifier outputs to find the kNN in the reference set. Then the classifiers that have correctly classified those neighbors are selected. Finally, the selected classifiers are combined in order to classify the input sample. This method is known as KNORA. The main drawback of KNORA is that a robust set of features must be defined in order to compute the similarity between the input sample and the samples in the dynamic

selection database.

A multi-hypothesis approach and classifier fusion is applied by Panton [32]. Each base classifier is constructed from a HMM that is trained from features extracted from local regions as well as from the signature as a whole. A distinct set of signatures with genuine and skilled forgery samples constitute a convenient optimization set that is used to select the most proficient ensemble. A signature, that is claimed to belong to a legitimate client (member of the general public), is therefore rejected or accepted based on the majority vote decision of the base classifiers within the most proficient ensemble.

An ensemble of classifiers based on graphometric features is utilized to improve the reliability of the classification by Bertolini et al. [44]. The ensemble is built using a standard genetic algorithm and different fitness functions are assessed to drive the search. Two different scenarios are considered in experiments. In the former, it is assumed that only genuine signatures and random forgeries are available to guide the search. In the latter it is assumed that simple and simulated forgeries also are available during the optimization of the ensemble. The pool of base classifiers is trained using only genuine signatures and random forgeries.

Score level combination is examined for offline signature verification by Prakash and Guru [110]. Classifiers of distance and orientation features are used individually and in combination. Distance features and orientation features individually provide 21.61% and 19.88% DERs on MCYT-75 corpus. Max fusion decreases the DER to 18.26%. Average fusion decreases the DER to 17.33% where the weights are fixed empirically.

Guest and Miguel-Hurtado apply majority voting classifier combination of 4 different features [46]. They achieve 19.86% EER improvement with 5 references, over the best individual feature type.

We combine the classifiers of the features introduced in Chapter 3 for user-dependent and user-independent (global) cases. Explicitly written, for a single query signature there are 7 score outputs obtained: HOG-Cartesian USVM (S_{u1}), HOG-Polar USVM (S_{u2}), SIFT USVM (S_{u3}), LBP-Cartesian USVM (S_{u4}), HOG-Cartesian GSVM (S_{g1}), HOG-Polar GSVM (S_{g2}), LBP-Cartesian GSVM (S_{g3}). A simple score level linear combination is used to obtain the final score

$$S_f = [S_{u1}\ S_{u2}\ S_{u3}\ S_{u4}\ S_{g1}\ S_{g2}\ S_{g3}] * [w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7]' \tag{4.4}$$

where the weight set is found empirically from a validation set.

Chapter 5

Experimental Evaluation

5.1 Dataset

GPDS-300, a publicly available subset of the GPDS-960 dataset [23] is used to evaluate the system performance. We use the subset GPDS-160 for testing, to be compatible with most of the recent works. Remaining 140 subjects are used for training (GSVMs) and verification issues. Each individual provides 24 genuine signature samples. A total of 30 practiced (skilled) forgery signatures, provided by 10 forgers, are collected for each individual. Before collecting skilled forgery signatures of a corresponding individual, a number of high resolution signature images were made available to forgers for practice. Genuine signatures are collected in a single session, where each subject is asked to sign his/her signature into a form with a preprinted grid containing two types of cells 5x3.5cm and 5.5x2.5cm, respectively. Prior to collecting skilled forgery signatures of a corresponding individual, a number of high resolution signature images were made available to forgers for practice. Likewise, forgers submit corresponding signatures using forms with the similar grid size. Finally, both reference and skilled forgery signatures are scanned at 300dpi resolution and preprocessed to a black and white format. Figure 2.1 depicts sample genuine (first three columns) and their corresponding skilled forgery (last column) signatures from the dataset. A gray-level version of the database is also available, which is currently undisclosed for the public. Because of the difficulties and privacy issues in collecting a signature database, a synthetic offline signature generation method is introduced by Ferrer et al. [111].

5.2 Test Protocol

In order to obtain results that are comparable to those reported in the literature, we train classifiers using 12 reference signatures. However this many reference signatures are not common in real life applications. So, in the next part of our tests, we use 5 references to obtain results that better reflect applications where users are willing to provide only a few reference signatures for enrollment.

In skilled forgery tests, we use all genuine signatures of a user except those that are used as reference; thus resulting in 12 and 19 genuine tests per user, for the cases of 12 and 5 reference signatures, respectively. Since we do not use any skilled forgeries of test users in training, all skilled forgeries of a user (30) are used in testing. All errors are reported using EER. In Table 5.9 to compare with previous works that do not provide EER, we also report DER that is defined in Section 1.1.

5.2.1 Baseline System

We define a baseline system utilizing 5 genuine signatures as reference set to measure the improvement with particular contributions. Our baseline system utilizes LBP-1 features with a score level fusion of USVM and GSVM classifiers (namely LBP-1 USVM and LBP-1 GSVM). This baseline system is referred to as **baseline** where ever a particular result with the baseline system is reported.

5.3 Results

USVM results that we obtain are given in Table 5.1 and Table 5.2; GSVM results are given in Table 5.3 and different combination results are given in Table 5.4. Analysis of these results shows that the USVM significantly outperforms GSVM. This is not very surprising as the USVMs are specifically trained for each user, while GSVMs only know about global (across all users) variations in each dimension. On the other hand, the global SVM improves the performance when used in conjunction with user SVMs.

Classifier combination is applied at score level to combine the decisions of the six classifiers. As found in many studies in different fields, we also find that classifier

combination using a weighted sum rule improves overall accuracy (6.97% EER using 12 references and 7.98% EER using 5 references). The weights are coarsely found in a separate validation set using grid search.

Note that with the GSVM, training is done using a separate set of users. Also, since a model is trained globally, the number of individual reference signatures is arbitrary. With large features (features other than HOG in Cartesian coordinates), we use the same GSVM model (trained with 5 references) for all test cases. In test phase, difference vectors can be calculated using 5 or 12 references.

As for the features, LBP features outperform all other types of features with 8.75% EER using 12-reference USVMs. Because HOG feature in Cartesian coordinates is relatively compact, it is possible to train GSVM with 12 references, giving the best GSVM result of 20.55%. Finally, we observe that using a greater number of reference signatures significantly improves the performance, as expected and observed in all other previous works also.

As can be seen from the table, alignment improves the results of GSVM, for instance 2.54% in HOG-Grid GSVM with 5 references. Coarse to fine overlapping grids improved the performance of, for instance LBP0 features for 5 reference case, decreasing the EER up to 6.64%.

Features	Classification	12 ref.	5 ref.
HOG-Polar-Hierarchy	USVM	16.39%	18.26%
HOG-Grid-Hierarchy	USVM	19.54%	21.36%
SIFT-TH	USVM	20.51%	24.09%

Table 5.1: Summary of the EER performance results of genuine query and skilled forgery query tests for USVMs except LBP.

Features	Classification	12 ref.	5 ref.
LBP0-Global	USVM	21.30%	24.18%
LBP0-Grid-Hierarchy	USVM	15.32%	17.54%
LBP1-Grid-Hierarchy	USVM	15.01%	16.94%
LBP2-Grid-Hierarchy	USVM	15.10%	17.43%
$LBP2_{rnd}$-Grid-Hierarchy (average EER of experiments)	USVM	19.29%	21.45%
$LBP2_{min}$-Grid-Hierarchy	USVM	21.68%	22.43%
$LBP2_{n32}$-Grid-Hierarchy	USVM	15.78%	18.16%
LBP0F-Grid-Hierarchy	USVM	10.04%	11.17%
LBP1F-Grid-Hierarchy	USVM	11.34%	12.08%
LBP2F-Grid-Hierarchy	USVM	11.30%	12.77%
$LBP2F_{rnd}$-Grid-Hierarchy (average EER of experiments)	USVM	10.32%	11.42%
$LBP2F_{min}$-Grid-Hierarchy	USVM	9.64%	11.01%
$LBP2F_{n32}$-Grid-Hierarchy	USVM	11.16%	11.20%
Fusion(LBP2 variations)-Grid-Hierarchy	USVM	8.75%	9.13%

Table 5.2: Summary of the EER performance results of genuine query and skilled forgery query tests for LBP USVMs.

Features	Classification	12 ref.	5 ref.
HOG-Polar-Hierarchy	GSVM	24.00%	25.41%
HOG-Polar-Hierarchy	GSVM (aligned)	22.35%	23.87%
HOG-Grid-Hierarchy	GSVM	20.83%	26.13%
HOG-Grid-Hierarchy	GSVM (aligned)	20.55%	23.49%
$LBP2F_{5x5}$-Grid-Hierarchy (sample1)	GSVM	30.25%	30.32%
$LBP2F_{5x5}$-Grid-Hierarchy (sample1)	GSVM (aligned)	26.96%	26.84%

Table 5.3: Summary of the EER performance results of genuine query and skilled forgery query tests for GSVMs.

To compare different LBP farther neighborhood pattern selection schemes, de-

Features	Classification	12 ref.	5 ref.
Combi.	USVMs, LBP_{fusion} for LBP	7.84%	8.57%
Combi.	GSVMs (not aligned)	17.14%	20.60%
Combi.	GSVMs (aligned)	18.32%	20.88%
All	Combination with USVMs (LBP_{fusion} for LBP) and GSVMs (not aligned)	7.57%	8.38%
All	Combination with USVMs (LBP_{fusion} for LBP) and GSVMs (aligned)	7.57%	8.38%
All	Combination with USVMs (LBP_{fusion} for LBP) and GSVMs	7.32%	8.30%
All	Combined, more precise weights	6.97%	7.98%

Table 5.4: Summary of the EER performance results of genuine query and skilled forgery query tests for different combinations.

tailed results are provided for individual distance samples. LBP-0F and LBP-1F results are shown in Table 5.5, LBP-2F results are shown in Table 5.6, LBP-2F random pattern selection results are shown in Table 5.7. Classifier combination of different pattern selection schemes are provided in Table 5.8. It can be seen that even highest $|\Delta TF|$ value pattern selection-based LBP2F is more successful in individual distance-sample classifiers; score averaging classifier combination of several random pattern selection experiments and LBP2F_{rnd} final score fusion classifiers are better. This can be explained by the information carried out just with the highest $|\Delta TF|$ value-based selected fixed 32 patterns versus boost of random patterns. LBP2F_{min} is also more successful as final classifier. This is explained with the discriminative value of low-frequency patterns in final decision. Final score level combination of highest and lowest $|\Delta TF|$ value-based pattern selection is also investigated as well as with highest next 32 patterns, namely LBP2F_{n32}, yielding to EER values under 10%.

Applying alignment in USVM training phase (that is aligning the random forgeries to the first reference, taking just one reference from each other user as random forgery) decreases the EER roughly 0.25% on baseline system. However as random

Method	12 ref.	5 ref.
LBP0$F-Grid-Hierarchy_{3x3}$	15.32%	17.54%
LBP0$F-Grid-Hierarchy_{5x5}$ (average EER of samples 1&2)	11.21%	13.02%
LBP0$F-Grid-Hierarchy_{7x7}$ (average EER of samples 1&2&3)	10.85%	11.43%
LBP0$F-Grid-Hierarchy$ (score fusion of distances)	10.04%	11.22%
LBP1$F-Grid-Hierarchy_{3x3}$ (feat. combi. of 2 samples)	15.01%	16.94%
LBP1$F-Grid-Hierarchy_{5x5}$ (feat. combi. of 4 samples)	12.27%	13.37%
LBP1$F-Grid-Hierarchy_{7x7}$ (feat. combi. of 6 samples)	12.97%	13.30%
LBP1$F-Grid-Hierarchy_{9x9}$ (feat. combi. of 8 samples)	13.81%	13.98%
LBP1$F-Grid-Hierarchy$ (score fusion of distances)	11.34%	12.08%

Table 5.5: Detailed LBP farther neighborhood group results for LBP-0F and LBP-1F.

forgeries (negative training samples), we take 1 reference from each other user, because of computational complexity. We do not apply this scheme in our final system because of high cost and relatively low performance gain.

For comparison, we give recent state-of-the-art results on the GPDS database in Table 5.9. Compared to the results given in this table, our classifier combination result is better than all the other systems in the literature that incorporate the same experimental setup. This comparison set includes systems that use skilled forgeries in training [29, 40, 41] (we do not use any skilled forgeries in training), as well as systems that use simpler subsets of the GPDS database [3, 35, 37, 41, 42, 47] (we use GPDS-160). Some systems utilize the gray-level version of the database to study gray-level features, which is not applicable for our work [3, 40–42, 47]. Note that the use of user-dependent thresholds that assumes knowledge of a user‘s forgeries is not suitable for real life applications since it is not viable to have real forgeries.

Method	12 ref.	5 ref.
LBP2$F - Grid - Hierarchy_{3x3}$	15.10%	17.43%
LBP2$F - Grid - Hierarchy_{5x5}$ (average EER of samples 1&2)	11.99%	13.62%
LBP2$F - Grid - Hierarchy_{7x7}$ (average EER of samples 1&2&3)	12.18%	13.10%
LBP2$F - Grid - Hierarchy$(score fusion of distances)	11.30%	12.77%
LBP2$F_{n32} - Grid - Hierarchy_{3x3}$	15.78%	18.16%
LBP2$F_{n32} - Grid - Hierarchy_{5x5}$ (average EER of samples 1&2)	12.72%	13.49%
LBP2$F_{n32} - Grid - Hierarchy_{7x7}$ (average EER of samples 1&2&3)	13.84%	13.85%
LBP2$F_{n32} - Grid - Hierarchy$(score fusion of distances)	11.16%	11.20%
LBP2$F_{min} - Grid - Hierarchy_{3x3}$	21.68%	22.43%
LBP2$F_{min} - Grid - Hierarchy_{5x5}$ (average EER of samples 1&2)	17.54%	19.71%
LBP2$F_{min} - Grid - Hierarchy_{7x7}$ (average EER of samples 1&2&3)	13.72%	15.33%
LBP2$F_{min} - Grid - Hierarchy$(score fusion of distances)	9.64%	11.01%

Table 5.6: Detailed LBP farther neighborhood group results for LBP-2F, different pattern selection methods.

Method	12 ref.	5 ref.
LBP2$F_{rnd\&scorefusion(ofseveralexperiments)}$ – $Grid$ – $Hierarchy_{3x3}$	14.57%	19.56%
LBP2$F_{rnd\&scorefusion(ofseveralexperiments)}$ – $Grid$ – $Hierarchy_{5x5}$ (average EER of samples 1&2)	11.90%	12.77%
LBP2$F_{rnd\&scorefusion(ofseveralexperiments)}$ – $Grid$ – $Hierarchy_{7x7}$ (average EER of samples 1&2&3)	11.10%	11.69%
LBP2$F_{rnd\&scorefusion(ofseveralexperiments)}$ – $Grid$ – $Hierarchy$ (score fusion of distances)	9.89%	10.83%
LBP2$F_{rnd\&averageEER(ofseveralexperiments)}$ – $Grid$ – $Hierarchy_{3x3}$	19.29%	21.45%
LBP2$F_{rnd\&averageEER(ofseveralexperiments)}$ – $Grid$ – $Hierarchy_{5x5}$ (average EER of samples 1&2)	13.23%	14.54%
LBP2$F_{rnd\&averageEER(ofseveralexperiments)}$ – $Grid$ – $Hierarchy_{7x7}$ (average EER of samples 1&2&3)	12.84%	13.57%
LBP2$F_{rnd\&averageEER(ofseveralexperiments)}$ – $Grid$ – $Hierarchy$ (score fusion of distances)	10.32%	11.42%

Table 5.7: Detailed LBP farther neighborhood results for LBP-2F random pattern selection.

Method	12 ref.	5 ref.
Score fusion of final LBP2 variations	9.65%	9.96%
Score fusion of final LBP2 variations, more precise weights	8.75%	9.13%

Table 5.8: LBP farther neighborhood results for combination of individual pattern selections.

Reference	Method	GPDS Set	Training	Testing	DER
Ferrer et. al. [29]	HMM	GPDS-160	12 gen. + **3 skl. forg.**	12 gen. + 27 skl. forg.	13.35%
Nguyen et. al. [30]	SVM	GPDS-160	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	20.07%
Vargas et. al. [40]	PNN	GPDS-160 **gray level**	12 gen. + **12 skl. forg.**	12 gen. + 12 skl. forg.	12.33%
Larkins and Mayo [37]	Adaptive Feature Thresholding	**GPDS-39**	12 gen. ref. + rand. forg. (gen. ref. from other users)	12 gen. and 30 skl. forg. for each user	14.01%
Vargas et. al. [41]	Pseudo-cepstral coefficients	**GPDS-100 gray level**	12 gen. + **12 skl. forg.**	12 gen. + 12 skl. forg.	6.20%
Nguyen et. al. [21]	Global feat.	GPDS-160	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	17.25%
Vargas et. al. [42]	Wavelets	**GPDS-100 gray level**	5 gen. + rand. forg.	19 gen. and **24 skl. forg.** for each user	14.22%
Vargas et. al. [3]	Texture feat.	**GPDS-100 gray level**	10 gen. + rand. forg.	14 gen. and **24 skl. forg.** for each user	9.02%

Parodi et. al. [45]	Graphometric feat.	**$GPDS_{rand}130$**	**13 gen.** + rand. forg.	11 gen. and **24 skl. forg., simple forg. (not detailed)** for each user	4.21%
Batista et. al. [59]	EoCs	GPDS-160	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	16.81%
Bharathi and Shekar [35]	Chain code histogram	**GPDS-100**	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	11.4%
Eskander et. al. [63]	Writer indep. classifier	GPDS-160	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	26.73%
Eskander et. al. [63]	Writer dependent classifier	GPDS-160	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	22.71%
Hu and Chen [47]	Writer dependent SVM	**$GPDS_{rand}150$ gray level**	10 gen. + rand. forg.	14 gen. + 30 skl. forg.	7.66%
Hu and Chen [47]	Writer indep. Adaboost	**$GPDS_{rand}100$ gray level**	10 gen. + rand. forg.	14 gen. + 30 skl. forg.	9.94%
Proposed	SVM	GPDS-160	12 gen. + rand. forg.	12 gen. + 30 skl. forg.	6.97%
Proposed	SVM	GPDS-160	5 gen. + rand. forg.	19 gen. + 30 skl. forg.	7.98%

Table 5.9: Summary of recent results (DER) on GPDS dataset.

5.3.1 Effect of Varying Reference Sets

In order to investigate the statistical significance of the reported results, we tried three different reference sets for a limited number of configurations, due to extensive training times. If N is chosen as 12, then first 12 genuine signatures are selected as the reference set for each user by default, for simplicity and consistency. This type of reference set selection is referred as R_1. As another choice, last 12 genuine signatures are selected as reference set and this reference set selection is referred as R_2. As the last choice, genuine signatures with indices [7-18] are selected as reference set and referred as R_3. Results obtained with various configurations and reference set selections are provided in Table 5.10.

As can be seen in this table, there is roughly a 2.5 standard deviation in the EER obtained with different reference sets, while the EER rates themselves range between 14-25%. While this is relatively high standard deviation, we note that the rank of the three methods remain unchanged: best is LBP2 (2nd line) and worst is HOG with GSVM (3rd line).

Except for this table, the first N genuine signatures are chosen as the reference set in all of our experiments, where N is the number of references.

Method	R_1 EER	R_2 EER	R_3 EER	Std. dev.
HOG-Polar-Hierarchy USVM	16.39%	19.75%	15.00%	2.44
LBP2-Grid-Hierarchy USVM	15.10%	19.07%	13.79%	2.75
HOG-Grid-Hierarchy aligned GSVM	20.55%	24.90%	20.02%	2.68

Table 5.10: Effect of varying reference sets.

5.3.2 Effect of Varying the Number of References for GSVMs

GSVMs allow any number of references during testing, as the data is pooled across all users. We investigate the effect of varying the number of references for GSVMs with a single case of HOG-Grid-Hierarchy aligned-GSVM trained with 5 references. Considered quantity of test references varies from 1 to 16. Plot of number of references versus EER is shown in Figure 5.1.

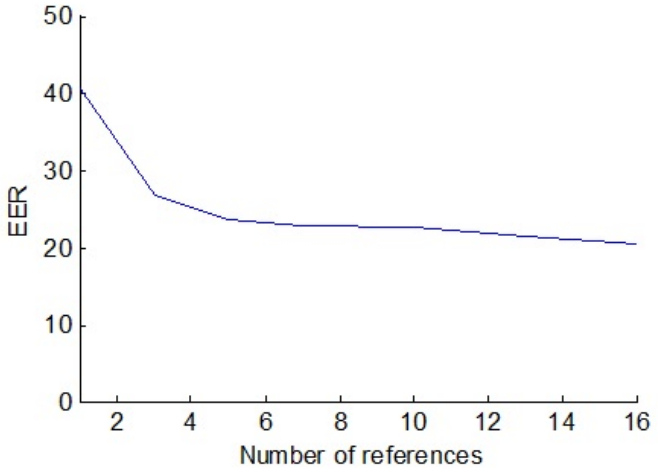


Figure 5.1: Effect of varying the number of references for HOG-Grid-Hierarchy aligned-GSVM.

5.4 Running Times

Running times of different modules of the verification system are measured. Codes are not optimized with any concern of speed improvement. Codes are run mostly interpreted but not compiled; on a PC with quad core 2 GHz CPU with 6 MB cache, 6 GB system memory and 64 bit operating system.

Signature preprocessing times are reported in Table 5.11, feature extraction times are reported in Table 5.12, classifier training times are reported in Table 5.13 and classifier testing times are reported in Table 5.14.

Operation	Average time (per signature) in sec.
Image preprocessing	3.00×10^{-2}
Applying all transformations to one reference	2.60×10^{0}
Alignment of a query to one reference	8.93×10^{-2}

Table 5.11: Running times of signature preprocessing operations.

Operation	Average time (per signature) in sec.
HOG-Polar feature extraction	5.70×10^{-1}
HOG-Grid feature extraction	4.00×10^{-2}
LBP0 feature extraction	1.00×10^{-2}
LBP1-Grid feature extraction	1.00×10^{-1}
LBP2-Grid feature extraction	1.30×10^{-1}
LBP0F-Grid feature extraction	2.54×10^{0}
LBP1F-Grid feature extraction	1.18×10^{0}
LBP2F-Grid feature extraction	7.80×10^{-1}
SIFT-TH feature extraction	9.10×10^{-1}

Table 5.12: Running times of feature extraction operations.

Operation	Average time (per user (USVM) or all training users (GSVM)) in sec.
HOG-Polar 5 ref. USVM training per user	2.71×10^{1}
HOG-Grid 5 ref. USVM training per user	2.90×10^{-1}
LBP0 5 ref. USVM training per user	3.00×10^{-2}
LBP1-Grid 5 ref. USVM training per user	5.30×10^{-1}
LBP2-Grid 5 ref. USVM training per user	5.30×10^{-1}
LBP0F-Grid 5 ref. USVM training per user	2.84×10^{1}
LBP1F-Grid 5 ref. USVM training per user	1.95×10^{1}
LBP2F-Grid 5 ref. USVM training per user	3.33×10^{0}
SIFT-TH 5 ref. USVM training per user	3.80×10^{-1}
HOG-Polar 5 ref. GSVM training	2.43×10^{4}
HOG-Grid 5 ref. GSVM training	7.92×10^{3}
LBP2$F_{5x5-sample1}$-Grid 5 ref. GSVM training	2.83×10^{4}

Table 5.13: Running times of classifier training operations.

Operation	Average time (per signature) in sec.
HOG-Polar 5 ref. USVM testing	1.00×10^{-2}
HOG-Grid 5 ref. USVM testing	1.00×10^{-4}
LBP0 5 ref. USVM testing	2.00×10^{-5}
LBP1-Grid 5 ref. USVM testing	2.00×10^{-4}
LBP2-Grid 5 ref. USVM testing	1.00×10^{-4}
LBP0F-Grid 5 ref. USVM testing	1.35×10^{-2}
LBP1F-Grid 5 ref. USVM testing	5.80×10^{-3}
LBP2F-Grid 5 ref. USVM testing	9.00×10^{-4}
SIFT-TH 5 ref. USVM testing	7.00×10^{-5}
HOG-Polar 5 ref. GSVM testing	6.10×10^{-1}
HOG-Grid 5 ref. GSVM testing	2.50×10^{-1}
LBP2$F_{5x5-sample1}$-Grid 5 ref. GSVM testing	7.54×10^{-1}

Table 5.14: Running times of classifier testing operations.

Chapter 6

Conclusions

We present an automatic offline signature verification system based on signature's local histogram representations. The signature is divided into zones using both fixed size rectangular or polar grids, where HOG and LBP features are calculated. For either of the representations, features obtained from grid zones are concatenated to form the final feature vector. Two different types of SVM classifiers are trained, namely global and user-dependent SVMs, to perform verification. We also experiment with the fusion of classifiers, and show that the combination improves overall verification performance. Feature-level fusion is possible but we prefer to train classifiers to be experts for each feature type. Score-level fusion helps the classification process with an additional information on how to combine different feature classifiers by deciding on combination weights, separately.

As stated earlier, results depend on the database (GPDS-100, GPDS-160), existence of skilled forgeries in training and testing, and image type (binary or gray-level). For an overall comparison, best previous result on binary GPDS-160 with 12 genuine signatures as reference set and without skilled forgeries in training is reported as 16.81% DER [59]. Works that use skilled forgeries in training report better results but they are not applicable in real-life scenarios; such as the work by Ferrer et al. [29] which uses 12 skilled forgeries and 12 genuine signatures in training, reporting 12.33% DER. Vargas et al. [40] use 3 skilled forgeries and 12 genuine signatures per user resulting in 13.35% DER.

Our system performance is measured using genuine query and skilled forgery query tests on the GPDS-160 signature dataset. Additionally, a classifier fusion is performed, where global and user-dependent SVM classifiers are combined giving the

best result of 6.97% and 7.98% EERs with 12 and 5 genuine references, respectively.

In summary, obtained results are comparable or better compared to those reported in the literature for the GPDS database. Considering that using skilled forgeries brings a potentially significant advantage in accuracy, the results should be deemed comparable and possibly better than state-of-the-art results. On the other hand, the fact that the proposed system does not require skilled forgeries of the enrolling user, is attractive for real life applications.

Future work: While state-of-art in offline signature verification achieves around 10-15% EER in various databases, the performance of these systems would be expected to be significantly worse with signatures collected in real life scenarios. In the future, systems research needs to concentrate on increasing the robustness of systems towards larger variations encountered in real life. For instance signatures signed in smaller spaces, or in a hurry, or on documents with interfering lines.

Another issue is to allow the system work well with few number of references, such as three as is the case in many banking operations or even with one reference. Importance of user-based score normalization becomes significant with such extreme cases. Developing a simpler and better score normalization method is a part of our future work.

Measuring the complexity level of a signature can help with many issues such as user-based score normalization or security enforcement.

We plan to add complementary features such as sparse dictionary codebooks and gradient magnitudes in addition to gradient directions, as well as work on the above mentioned issues of robust features, signature normalization, and signature complexity analysis for user-based score normalization among other things.

Bibliography

[1] "b2bedocuments - biometric signature," http://www.b2bedocuments.com/html/biometricsignature02.htm.

[2] F. Alonso-Fernandeza, M. C. Fairhurstb, J. Fierrez, and J. Ortega-Garcia, "Impact of signature legibility and signature type in off-line signature verification," in *Biometrics Symposium 2007*. IEEE, September 2007, pp. 1–6.

[3] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-line signature verification based on grey level information using texture features," *Pattern Recogn.*, vol. 44, pp. 375–385, February 2011.

[4] G. S. Spagnolo, C. Simonetti, and L. Cozzella, "Superposed strokes analysis by conoscopic holography as an aid for a handwriting expert," *Journal of Optics A: Pure and Applied Optics*, vol. 6, pp. 869–874, 2004.

[5] A. Kholmatov and B. Yanıkoğlu, "Identity authentication using improved online signature verification method," *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2400–2408, 2005.

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

[7] R. Sabourin, R. Plamondon, and G. Lorette, "Off-line identification with handwritten signature images: survey and perspectives," *Structured Document Image Analysis*, pp. 219–234, 1992.

[8] F. Leclerc and R. Plamondon, "Automatic signature verification: the state of the art, 1989–1993," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 3, pp. 643–660, 1994.

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

[10] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *Trans. Sys. Man Cyber Part C*, vol. 38, no. 5, pp. 609–635, Sep. 2008.

[11] M. S. Arya and V. S. Inamdar, "A preliminary study on various off-line hand written signature verification approaches," *International Journal of Computer Applications*, vol. 1, no. 9, pp. 55–60, February 2010, published By Foundation of Computer Science.

[12] S. Pal, M. Blumenstein, and U. Pal, "Off-line signature verification systems: a survey," in *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, ser. ICWET '11. New York, NY, USA: ACM, 2011, pp. 652–657.

[13] V. Bhosale and A. Karwankar, "Automatic static signature verification systems: A review," *International Journal Of Computational Engineering Research*, vol. 3, no. 2, pp. 8–12, February 2013.

[14] D. Impedovo, G. Pirlo, and M. Russo, "Recent advances in offline signature identification," in *Frontiers in Handwriting Recognition, 2014 14th International Conference on*. IEEE, 2014.

[15] J. Coetzer, "Off-line signature verification," Ph.D. dissertation, University of Stellenbosch, South Africa, 2005.

[16] V. Bouletreau, N. Vincent, R. Sabourin, and H. Emptoz, "Handwriting and signature: one or two personality identifiers?" in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2, Aug 1998, pp. 1758–1760 vol.2.

[17] A. Chalechale, G. Naghdy, P. Premaratne, and A. Mertins, "Cursive signature extraction and verification," in *Proc. 2nd Int. Workshop on Information Technology & Its Disciplines (WITID 2004)*, Kish Island, Iran, July 2004, pp. 109–113.

[18] İ. Cüceloğlu and H. Oğul, "Detecting handwritten signatures in scanned documents," in *Computer Vision Winter Workshop*, 2014.

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

[20] S. Pal, A. Alireza, U. Pal, and M. Blumenstein, "Multi-script off-line signature identification," in *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, Dec 2012, pp. 236–240.

[21] V. Nguyen, M. Blumenstein, and G. Leedham, "Global features for the off-line signature verification problem," in *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ser. ICDAR '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1300–1304.

[22] M. A. Ferrer, F. Vargas, A. Morales, and A. Ordonez, "Robustness of offline signature verification based on gray level features." *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 966–977, 2012.

[23] F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-line handwritten signature GPDS-960 corpus," in *IAPR 9'th International Conference on Document Analysis and Recognition*, September 2007, pp. 764–768.

[24] G. Ganapathi and N. Rethinaswamy, "A fuzzy framework for offline signature verification," in *Electronics, Computing and Communication Technologies (IEEE CONECCT), 2014 IEEE International Conference on*, Jan 2014, pp. 1–6.

[25] "CEDAR signature database," http://www.cedar.buffalo.edu/Databases.

[26] R. Sabourin, G. Genest, and F. J. Preteux, "Off-line signature verification by local granulometric size distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 976–988, 1997.

[27] L. Bastos, F. Bortolozzi, R. Sabourin, and C. Kaestner, "Mathematical modelation of handwritten signatures by conics," *Revista da Sociedade Paranaese de Matematica*, 1997.

[28] J. K. Guo, "Forgery detection by local correspondence," Ph.D. dissertation, College Park, MD, USA, 2000, director-Rosenfeld, Azriel.

[29] M. A. Ferrer, J. B. Alonso, and C. M. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 993–997, 2005.

[30] V. Nguyen, M. Blumenstein, V. Muthukkumarasamy, and G. Leedham, "Off-line signature verification using enhanced modified direction features in conjunction with neural classifiers and support vector machines," in *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 734–738.

[31] J. Coetzer, B. M. Herbst, and J. A. du Preez, "Offline signature verification using the discrete radon transform and a hidden markov model," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 559–571, 2004.

[32] M. S. Panton, "Off-line signature verification using ensembles of local radon transform-based HMMs," Master's thesis, University of Stellenbosch, South Africa, 2010.

[33] M. Hanmandlu, M. Yusof, and V. Madasu, "Off-line signature verification and forgery detection using fuzzy modeling," *Pattern Recognition*, vol. 38, pp. 341–356, 2005.

[34] J. J. Igarza, I. Hernez, I. Goirizelaia, K. Espinosa, and J. Escolar, "Off-line signature recognition based on dynamic methods," in *Proceedings of the SPIE*, 2005, pp. 336–343.

[35] R. Bharathi and B. Shekar, "Off-line signature verification based on chain code histogram and support vector machine," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, Aug 2013, pp. 2063–2068.

[36] A. Gilperez, F. Alonso-Fernandez, S. Pecharroman, J. Fierrez, and J. Ortega-Garcia, "Off-line signature verification using contour features," in *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, ICFHR 2008*, Aug. 2008.

[37] R. Larkins and M. Mayo, "Adaptive feature thresholding for off-line signature verification," in *IVCNZ '08: Proceedings of the 23rd International Conference In Image and Vision Computing New Zealand*, 2008, pp. 1–6.

[38] J. Ruiz-Del-Solar, C. Devia, P. Loncomilla, and F. Concha, "Offline signature verification using local interest points and descriptors," in *CIARP '08: Proceedings of the 13th Iberoamerican congress on Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 22–29.

[39] 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, 2014 14th International Conference on*. IEEE, 2014, pp. 621–626.

[40] F. J. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-line signature verification based on high pressure polar distribution," in *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, ICFHR 2008*, August 2008, pp. 373–378.

[41] J. F. V. Bonilla, M. A. F. Ballester, C. M. T. Gonzalez, and J. B. A. Hernandez, "Offline signature verification based on pseudo-cepstral coefficients," in *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ser. ICDAR '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 126–130.

[42] J. Vargas, C. Travieso, J. Alonso, and M. A. Ferrer, "Off-line signature verification based on gray level information using wavelet transform and texture

features," in *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, Nov 2010, pp. 587–592.

[43] B. Zhang, "Off-line signature verification and identification by pyramid histogram of oriented gradients," *International Journal of Intelligent Computing and Cybernetics*, vol. 3, pp. 611–630, 2010.

[44] 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, pp. 387–396, 2010.

[45] M. Parodi, J. C. Gomez, and A. Belaid, "A circular grid-based rotation invariant feature extraction approach for off-line signature verification." in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 1289–1293.

[46] R. Guest and O. Miguel-Hurtado, "Enhancing off-line biometric signature verification using a fingerprint assessment approach," in *Proceedings of the 2011 IEEE International Carnahan Conference on Security Technology (ICCST)*, 2011.

[47] J. Hu and Y. Chen, "Offline signature verification using real adaboost classifier combination of pseudo-dynamic features," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 1345–1349.

[48] R. Wajid and A. Bin Mansoor, "Classifier performance evaluation for offline signature verification using local binary patterns," in *Visual Information Processing (EUVIP), 2013 4th European Workshop on*, June 2013, pp. 250–254.

[49] B. Ribeiro, I. Gonçalves, 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*, ser. Lecture Notes in Computer Science, C. San Martin and S.-W. Kim, Eds. Springer Berlin Heidelberg, 2011, vol. 7042, pp. 523–532.

[50] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Persian signature verification using convolutional neural networks," *International Journal of Engineering Research & Technology*, vol. 1, 2012.

[51] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[52] E. N. Zois, E. Zervas, K. Barkoula, G. Economou, and S. Fotopoulos, "Poset description of grid features and application to off-line signature verification," in *Frontiers in Handwriting Recognition, 2014 14th International Conference on*. IEEE, 2014.

[53] I. S. I. ABUHAIBA, "Offline signature verification using graph matching," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 15, no. 1, p. 89, March 2007.

[54] A. Piyush Shanker and A. N. Rajagopalan, "Off-line signature verification using DTW," *Pattern Recogn. Lett.*, vol. 28, no. 12, pp. 1407–1414, 2007.

[55] J. F. Velez, A. Sanchez, A. B. Moreno, and L. Morillo-Velarde, "Comparing elastic alignment algorithms for the off-line signature verification problem." in *IWINAC (2)*, ser. Lecture Notes in Computer Science, J. M. Ferrndez, J. R. l. Snchez, F. de la Paz, and F. J. Toledo, Eds., vol. 6687. Springer, 2011, pp. 233–242.

[56] W. Tian and J. Lv, "A new affine registration algorithm applied to off-line signature verification," in *Information and Automation (ICIA), 2012 International Conference on*, June 2012, pp. 806–810.

[57] X. Xiao and G. Leedham, "Signature verification by neural networks with selective attention," *Applied Intelligence*, vol. 11, no. 2, pp. 213–223, 1999.

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

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

[60] H. Coetzer and R. Sabourin, "A human-centric off-line signature verification system," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 1, Sept 2007, pp. 153–157.

[61] Y. Guerbai, Y. Chibani, and N. Abbas, "One-class versus bi-class SVM classifier for off-line signature verification," in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, May 2012, pp. 206–210.

[62] C. Santos, E. J. R. Justino, F. Bortolozzi, and R. Sabourin, "An off-line signature verification method based on the questioned document expert's approach and a neural network classifier," in *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, Oct 2004, pp. 498–502.

[63] G. S. Eskander, R. Sabourin, and E. Granger, "Hybrid writer-independent writer-dependent offline signature verification system," *IET Biometrics*, vol. 2, pp. 169–181(12), December 2013.

[64] E. Özgündüz, T. Şentürk, and M. E. Karslıgil, "Off-line signature verification and recognition by support vector machine," in *in Proc. European Signal Processing Conference*, 2005.

[65] Y. Qiao, J. Liu, and X. Tang, "Offline signature verification using online handwriting registration," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 1–8, 2007.

[66] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll, "SVC2004: First international signature verification competition," in *Biometric Authentication*, ser. Lecture Notes in Computer Science, D. Zhang and A. Jain, Eds. Springer Berlin Heidelberg, 2004, vol. 3072, pp. 16–22.

[67] B. Rabil, R. Sabourin, and E. Granger, "Impact of watermarking on offline signature verification in intelligent bio-watermarking systems," in *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2011 IEEE Workshop on*, April 2011, pp. 13–20.

[68] G. S. Eskander, R. Sabourin, and E. Granger, "A bio-cryptographic system based on offline signature images." *Inf. Sci.*, vol. 259, pp. 170–191, 2014.

[69] ——, "Improving signature-based biometric cryptosystems using cascaded SV-FV approach," in *Frontiers in Handwriting Recognition, 2014 14th International Conference on*. IEEE, 2014, pp. 187–192.

[70] "Caltech signature database," http://www.vision.caltech.edu/mariomu.

[71] "HIT–MW chinese signature database," https://sites.google.com/site/hitmwdb.

[72] C. Freitas, M. Morita, L. Oliveira, E. Justino, A. Yacoubi, E. Lethelier, F. Bortolozzi, and R. Sabourin, "Bases de dados de cheques bancarios brasileiros," in *XXVI Conferencia Latinoamericana de Informatica*, 2000.

[73] M. Liwicki, C. van den Heuvel, B. Found, and M. Malik, "Forensic signature verification competition 4NSigComp2010 - detection of simulated and disguised signatures," in *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, Nov 2010, pp. 715–720.

[74] M. Liwicki, M. Blumenstein, E. van den Heuvel, C. Berger, R. Stoel, B. Found, X. Chen, and M. Malik, "SigComp11: Signature verification competition for on- and offline skilled forgeries," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 2011.

[75] M. Liwicki, M. Malik, L. Alewijnse, E. van den Heuvel, and B. Found, "ICFHR 2012 competition on automatic forensic signature verification (4NSigComp2012)," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, Sept 2012, pp. 823–828.

[76] M. Malik, M. Liwicki, L. Alewijnse, W. Ohyama, M. Blumenstein, and B. Found, "ICDAR 2013 competitions on signature verification and writer

identification for on- and offline skilled forgeries (SigWiComp2013)," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 1477–1483.

[77] M. K. Kalera, S. Srihari, and A. Xu, "Off-line signature verification and identification using distance statistics," in *International Journal of Pattern Recognition and Artificial Intelligence*, 2003, pp. 228–232.

[78] X. You, B. Fang, Z. He, and Y. Tang, "Similarity measurement for off-line signature verification," in *Intelligent Computing, 2005 International Conference on*, vol. 3644, China, August 2005, pp. 272–281.

[79] M. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Transactions on images processing*, vol. 14, no. 12, pp. 2091–2106, December 2005.

[80] 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 International Conference on*. IEEE, 2009.

[81] "Graphometric features," in *Encyclopedia of Biometrics*, S. Li and A. Jain, Eds. Springer US, 2009, pp. 666–666.

[82] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

[83] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," 1994, pp. A:582–585.

[84] A. Porebski, N. Vandenbroucke, and D. Hamad, "LBP histogram selection for supervised color texture classification," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sept 2013, pp. 3239–3243.

[85] B. Sujatha, V. V. Kumar, and P. Harini, "A new logical compact LBP co-occurrence matrix for texture analysis," *International Journal of Scientific & Engineering Research*, 2012.

[86] T. Mäenpää and M. Pietikäinen, "Multi-scale binary patterns for texture analysis," in *Image Analysis*, ser. Lecture Notes in Computer Science, J. Bigun and T. Gustavsson, Eds. Springer Berlin Heidelberg, 2003, vol. 2749, pp. 885–892.

[87] X. Qi, Y. Qiao, C.-G. Li, and J. Guo, "Multi-scale joint encoding of local binary patterns for texture and material classification," in *British Machine Vision Conference (BMVC)*, Sept 2013.

[88] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, "Face detection based on multi-block LBP representation," in *Advances in Biometrics*, ser. Lecture Notes in Computer Science, S.-W. Lee and S. Li, Eds. Springer Berlin Heidelberg, 2007, vol. 4642, pp. 11–18.

[89] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[90] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[91] K. E. Mwangi, "Offline handwritten signature verification using SIFT features," Master's thesis, Makerere University, Uganda, 2008.

[92] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, vol. 3951, pp. 404–417.

[93] "SURF," http://en.wikipedia.org/wiki/SURF.

[94] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *submitted to IEEE Trans. Image Processing*, 2014.

[95] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[96] C. N. Duong, K. G. Quach, and T. D. Bui, "Are sparse representation and dictionary learning good for handwritten character recognition?" in *Frontiers in Handwriting Recognition, 2014 14th International Conference on.* IEEE, 2014.

[97] X. Wang, T. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 32–39.

[98] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[99] K. Barkoula, E. N. Zois, E. Zervas, and G. Economou, "Off-line signature verification based on ordered grid features: An evaluation." in *AFHA*, ser. CEUR Workshop Proceedings, M. I. Malik, M. Liwicki, L. Alewijnse, M. Blumenstein, C. Berger, R. Stoel, and B. Found, Eds., vol. 1022. CEUR-WS.org, 2013, pp. 36–40.

[100] M. B. Yılmaz, B. Yanıkoğlu, Ç. Tırkaz, and A. A. Kholmatov, "Offline signature verification using classifier combination of HOG and LBP features," in *Proc. of the 2011 Intl. Joint Conf. on Biometrics (IJCB 2011)*, ser. IJCB '11, Washington, DC, USA, 2011.

[101] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, pp. 121–167, June 1998.

[102] J. Fierrez-Aguilar, D. Garcia-Romero, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Adapted user-dependent multimodal biometric authentication ex-

ploiting general information," *Pattern Recogn. Lett.*, vol. 26, pp. 2628–2639, December 2005.

[103] G. S. Eskander, R. Sabourin, and E. Granger, "Adaptation of writer-independent systems for offline signature verification," in *Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition*, ser. ICFHR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 434–439.

[104] N. A. Murshed, F. Bortolozzi, and R. Sabourin, "Off-line signature verification, without a priori knowledge of class w2. a new approach," in *Document Analysis and Recognition, 1995., Proceedings of the Third IAPR Conference on*, 1995, pp. 191–196.

[105] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Target dependent score normalization techniques and their application to signature verification." in *ICBA*, ser. Lecture Notes in Computer Science, D. Zhang and A. K. Jain, Eds., vol. 3072. Springer, 2004, pp. 498–504.

[106] R. Sabourin and G. Genest, "An extended-shadow-code-based approach for off-line signature verification. ii. evaluation of several multi-classifier combination strategies," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1, Aug 1995, pp. 197–201 vol.1.

[107] L. S. Oliveira, E. Justino, R. Sabourin, and F. Bortolozzi, "Combining classifiers in the ROC-space for off-line signature verification," *Journal of Universal Computer Science*, vol. 14, no. 2, pp. 237–251, 2008.

[108] L. Batista, E. Granger, and R. Sabourin, "A multi-classifier system for off-line signature verification based on dissimilarity representation," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, N. El Gayar, J. Kittler, and F. Roli, Eds. Springer Berlin Heidelberg, 2010, vol. 5997, pp. 264–273.

[109] ——, "Dynamic ensemble selection for off-line signature verification," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, C. Sansone,

J. Kittler, and F. Roli, Eds. Springer Berlin Heidelberg, 2011, vol. 6713, pp. 157–166.

[110] H. N. Prakash and D. S. Guru, "Offline signature verification: An approach based on score level fusion," *International Journal of Computer Applications*, vol. 1, no. 18, pp. 52–58, 2010.

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