SIGNATURE VERIFICATION

Final Year Project Report
by

Muhammad Saifullah Khan
Muhammad Mahad Tariq
Bilal Ahmad

In Partial Fulfillment
of the requirements for the degree
Bachelors of Engineering in Software Engineering (BESE)

School of Electrical Engineering and Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
(2018)

DECLARATION

We hereby declare that this project report entitled "SIGNATURE VERIFICATION" submitted to the "DEPARTMENT OF SOFTWARE ENGINEERING", is a record of an original work done by us under the guidance of Supervisor "DR. MUHAMMAD IMRAN MALIK" and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Software Engineering.

Team Members	**Signature**
Muhammad Saifullah Khan	_____
Muhammad Mahad Tariq	_____
Bilal Ahmad	_____

Supervisor:	**Signature**
Dr. Muhammad Imran Malik	_____

Date: _____

Place: _____

DEDICATION

This thesis is dedicated to our parents for their love and prayers, and to our friends for their constant support.

ACKNOWLEDGEMENT

First of all, we would like to thank all members of our project committee for their guidance and encouragement throughout the course of this project, without which we might not have been able to accomplish this project. We would like to especially thank our project supervisor, Dr. Muhammad Imran Malik, whose patience and support was instrumental in overcoming all the many challenges that we faced in our research.

We are indebted to our department and the TUKL-NUST Research & Development Center for providing us with valuable resources which helped us conduct our research.

We are also grateful to our fellow students for their feedback and cooperation which allowed us to complete and fine-tune our project. In addition, we would also like to express gratitude to our families and friends whose support kept our morale up and helped us successfully conclude this project.

In the end, we would like to acknowledge the importance of Google Scholars, StackOverflow, IEEE Xplore, Tutorialspoint, and GitHub, without which any meaningful research or successful implementation would have been impossible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Signature verification is the process of using machine learning methods to validate the authenticity of an individual's signature. Signatures can be of one of the two types; on-line or off-line, and this project focuses on off-line signature verification. Aim of this project is to design an algorithm which can distinguish between genuine and forged signatures using writer independent features, and to develop a system using this algorithm which can be used to verify signatures on bank cheques. We intend to build a complete end-to-end hardware/software system which can be used to acquire signatures from bank cheques, perform signature verification, and display the results. For this purpose, various deep learning techniques were developed and tested on standard datasets for off-line signature verification, as well as on a dataset collected by ourselves.

Chapter 1

INTRODUCTION

1.1 AUTOMATIC SIGNATURE VERIFICATION

Automatic Signature Verification is the process of determining the authenticity of a signature using automated techniques, given a signature and identity of the alleged author of the signature. It involves comparing an unknown signature sample with known signature samples from the author, and determining whether the unknown sample originated from the same author or was forged by a different author.

1.2 TYPES OF SIGNATURES

Signature is a biometric which is used extensively for personal identification, because unlike other biometrics such as fingerprint or iris, people are usually more willing to share their signatures [1]. Based on the acquisition method, signatures are grouped into following two categories:

1.2.1 Online Signatures

Online signatures are acquired using a special stylus and/or tablet which captures temporal information of the signature such as trajectory of the stylus and pressure at each instance. These are also called dynamic signatures

1.2.2 Offline Signatures

Offline signatures are acquired by scanning signature image from a regular paper written with a normal pen. These are also called static signatures.

1.3 PROBLEM STATEMENT

Due to additional information available with online signatures, automated systems for online signature verification yield better results in general [2].

However, because of their popularity, offline verification is an active research area with different techniques being used to improve its accuracy and generalization [3]. Signature is a behavioural biometric which requires active participation of user and is characterised by a behavioural trait that can be learnt and acquired over time rather than a physiological trait. No two signatures by same person are exactly identical, giving rise to intrapersonal variation. Signatures even change over time. Therefore, verification of offline signatures is not a trivial pattern recognition problem.

1.3.1 Unmet Need and Target Market

Forging signatures on bank cheques causes huge financial losses to banks annually. In the U.S. alone, estimates suggest that millions of cheques are forged annually causing losses of billions of dollars [4]. Cheque forgery can be done in a number of ways, a major one of which is by skillfully forging signatures of the cheque owner. Identity theft by forged signatures is a significant cause of breach of personal security which may lead to various personal and financial losses to both individuals and organisations.

Offline signature verification is an attractive area for research, which has been extensively studied for decades. However, hardly anyone has tried to put this research to active practical use in the real world. Automated end-to-end systems for offline signature verification are not widely used. Parascript and TIS are two of the only global companies providing solutions for this problem.

In Pakistan, no such automated techniques are employed, especially in the banking sector. This is the unmet need our project aims to solve. A system for automatically verifying authenticity of signatures on bank cheques has potential benefit for not only the banking sector, but for all organisations and individuals which perform transactions through bank cheques, by protecting them from fraud and identity theft.

Furthermore, offline signature verification techniques also have potential applications which range from forensic investigations to commonplace signature authentication. Therefore, research work aiming to improve the state-of-the-art techniques of offline signature verification will aid in enhancing forensic investigation and biometric authentication systems, as well as in bridging the gap between pattern recognition based techniques of verification and forensic science.

1.4 VERIFICATION PIPELINE FOR SIGNATURES ON BANK CHEQUES

Verification of signatures on bank cheques is a subset of offline signature verification. A digital image of the bank cheque is obtained through a scanning device. The signature on the cheque is compared with known signatures of the owner of that cheque to determine whether the cheque was really signed by its owner.

The phases of signature verification for bank cheques are briefly defined in the following subsections.

1.4.1 Cheque Preprocessing

The digital image of the scanned bank cheque is preprocessed to remove cheque background and other noise. The output is a binary image of the bank cheque having white background with black text on it.

1.4.2 Signature Extraction

In this phase, given the binary image of cheque, the actual signature is located and extracted from the whole cheque, discarding all other information including printed text, logos, and handwritten text other than the signature.

1.4.3 Analysis (Feature Extraction)

One we have the image of the actual signature, a set of features is extracted from it. These features are a quantifiable, unique description of the signature and

define its intrinsic characteristics which allow us to associate it with a particular author.

1.4.4 Comparison and Evaluation (Classification)

This phase involves comparing the feature vector obtained from the unknown signature sample with feature vectors of known signatures of the alleged author, and deciding whether the unknown signature belongs to this author or not.

1.5 DOCUMENT STRUCTURE

Chapter 2 discusses the Literature Review. We discuss prior work done by researchers in the field of signature extraction from bank cheques and documents in general. We also discuss existing research in the field of offline signature verification, the state-of-the-art, and the various techniques used for automatic signature verification by the research community in past four decades. Performance measures used to evaluate the accuracy and performance of signature verification systems are also discussed.

Chapter 3 provides the detailed Problem Definition. It explores the problem background which gives context to this project. Significance of the problem, how our project approaches it, and implications of our solution are also discussed.

Chapter 4 describes our Methodology. It talks about the details of our solution with regards to each of the phases of the verification pipeline described above.

Chapter 5 discusses the architecture of the system, and functionality of each component. Chapter 6 and 7 talk about implementation details, datasets used for training and testing, and the results obtained. Chapter 8 concludes the thesis and provides an outlook on future work in domain of automated offline signature verification.

Chapter 2

LITERATURE REVIEW

The key concept of signature verification is discriminating a genuine signature from a non-genuine signature. State-of-the-art pattern recognition based verification techniques have been demonstrated by [5] to be more consistent and, on average, more accurate as compared to manual signature verification by Forensic Handwriting Experts (FHEs), with average verification accuracy of 52% by FHEs and 68% by automated methods.

Signature pre-processing is generally a mandatory step which is used to remove noise from signature image to reduce bias in extracted features. Different pre-processing methods including smoothing, size normalization, pixel thinning (or thickening [6]), etc. have been proposed, and [7] shows that their effect on subsequent stages is significant. Concerns about the loss of information during pre-processing, and its impacts on feature extraction have also been voiced [8].

2.1 SIGNATURE EXTRACTION

One of the most important tasks in automatic bank cheque processing is the extraction of handwritten signatures from bank cheques. In order to verify a signature it must be extracted accurately. The accuracy of signature verification depends upon how accurately it is extracted. Many solutions have been proposed for signature extraction and still there is need of better extraction technique in order to enhance the accuracy of signature verification. Following are some of the techniques used for signature extraction.

2.1.1 Sliding Window Technique

To segment signatures from bank cheques and other documents Madasu et al. [9], proposed an approach based on sliding window to calculate the entropy and finally fit the window to signature block.. In this, a sliding window is created

to move horizontally from left to right on the approximation area. The width of the window is fixed to a certain number of pixels and the height of the pixel will be set according to the height of the approximation area. As the sliding window moves by one pixel at a time, the density of the pixels within the current window is calculated. The entropy is a better choice than density because it introduces larger range of values leading to easier and more accurate segmentation. A major problem of this approach is that it is based on a priori information about the location of the signature.

2.2.2 Filiformity Criteria

Djeziri et al. proposed an approach in [10] to extract signatures from check backgrounds inspired from human visual perception. It is based on filiformity criteria whose specialized task is to extract lines. Based on filiformity measure, contour lines of objects are differentiated from handwritten lines on a local level. The local values provided by this measure are then processed by global thresholding, taking into account information about the whole image, to extract the signatures.

2.2.3 Multi-Scale Structural Saliency Map

In this method Zhu et al. [11] describe the structural saliency approach to signature detection that searches over range of scales $S = \sigma 1, \sigma 2, \cdots, \sigma n$. This approach select the initial scale $\sigma 1$ based on the resolution of the input image. It defines the multi-scale structural saliency for a curve Γ as

$$\Phi(\Gamma) = \max \sigma i \in S \; f(\Phi \sigma i \, (\Gamma \sigma i \,), \sigma i),$$

where $f : R\,2 \rightarrow R$ is a function that normalizes the saliency over its scale, and $\Gamma \sigma i$ is the obtained connected component corresponding to the curve at the scale σi . Using multiple scales for detection relaxes the requirement that the curve Γ be connected at a particular scale.

Detection at a particular scale σi proceeds in three steps. First, convolve the image with a Gaussian kernel Gσi , re-sample it using the Lanczos filter [12] at the factor dσi , and compute its edges using the Canny edge detector [13]. This is effectively obtaining a coarse representation of the original image in which small gaps in the curve are bridged by smoothing followed by re-sampling.

Next, form connected components on the edge image at scale σi , and compute the saliency of each component. Then, identify the most salient curves and use a grouping strategy based on proximity and curvilinear constraints to obtain the rest of the signature parts within their neighborhood.

2.3.4 Conditional Random Field

Mandal et al. [14] proposed an approach using conditional random field for segmentation of signatures from machine printed documents. This approach requires a large number of training samples to actually differentiate between printed text from signatures. In addition, the behavior of this approach in presence of logos and handwritten annotations are not reported.

2.2 FEATURE EXTRACTION

Feature extraction is arguably the most important stage of signature verification. Different kinds of features, including various global, local and geometric features [15], have been used by different researchers to yield different results, but no standardised feature set for every problem domain exists. Different ranking methods and classification algorithms for automatic feature selection have been proposed [16].

2.2.1 Local Features vs Global Features

There are two types of features, sometimes referred to as descriptors, are extracted from the images based on the application. They are local and global features. Global descriptors are generally used in image retrieval, object detection

and classification, while the local descriptors used for object recognition/identification.

Global features describe the image as a whole to the generalize the entire object whereas the local features describe the image patches (key points in the image) of an object. Global features include contour representations, shape descriptors, and texture features and local features represent the texture in an image patch. Shape Matrices [17], Invariant Moments [18], Histogram Oriented Gradients (HOG) and Co-HOG [19] are some examples of global descriptors. SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), LBP (Local Binary Patterns), Binary Robust Invariable Scalable Keypoints (BRISK), Maximally Stable Extremal Regions (MSER) and Fast Retina Keypoints (FREAK) are some examples of local descriptors [20][21][22][23][24].

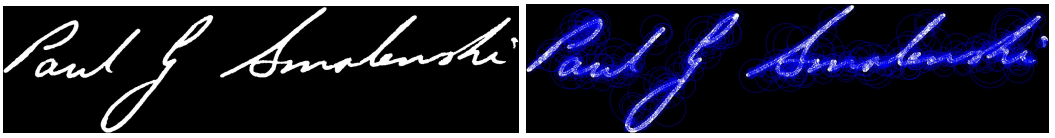


Fig 1(a): Original Signature Fig 1(b): SURF Features

Generally, for low level applications such as object detection and classification, global features are used and for higher level applications such as object recognition, local features are used. Combination of global and local features improves the accuracy of the recognition with the side effect of computational overheads.

2.2.2 Writer-Dependent vs Writer-Independent Features

The writer-dependent or personal model is based on one model per author. Usually it yields good results but its drawback lies in the fact that for each new author a new model should be built. Another important issue in this strategy is that usually a considerable amount of data is necessary to train a reliable model. It can be implemented using either one-against-all or pairwise strategy. This kind of approach has been largely used for signature verification [25].

An alternative to the personal approach is the global approach or writer-independent model. It is based on the forensic questioned document examination approach and classifies the writing, in terms of authenticity, into genuine and forgery, using for that one global model.

2.2.3 Handcrafted Feature Extractors

A large part of the research efforts on the field has been devoted to finding good feature representations for offline signatures [26][27]. Many descriptors have been 'handcrafted' for this purpose. Handcrafting refers to the fact that they were not found by a deep learning method but rather developed manually. In this section we summarize the main descriptors proposed for the problem.

1. **Geometric Features**

   Geometric features measure the overall shape of a signature. This includes basic descriptors, such as the signature height, width, caliber (height-to-width ratio) and area.

2. **Graphometric Features**

   Forensic document examiners use the concepts of graphology and graphometry to examine handwriting for several purposes, including detecting authenticity and forgery

3. **Directional Features**

   Directional features seek to describe the image in terms of the direction of the strokes in the signature.

4. **Mathematical Transformations**

   Researchers have used a variety of mathematical transformations as feature extractors such as Hadamard transform and spectrum analysis, Contourlet transform, Discrete Radon transform, Wavelet transform.

5. **Shadow-Code**

   Sabourin et al. proposed an Extended Shadow Code in [28] for signature verification. A grid is overlaid on top of the signature image, containing horizontal, vertical and diagonal bars, each bar containing a fixed number

of bins. Each pixel of the signature image is then projected to its closest bar in each direction, activating the respective bin. The count of active bins in the projections is then used as a descriptor of the signature.

6. **Texture Features**

   Texture features, in particular variants of Local Binary Patterns (LBP), have been used in many experiments in recent years. The LBP operator describe the local patterns in the image, and the histogram of these patterns is used as a feature descriptor.

7. **Interest Point Matching**

   Interest point matching methods, such as SIFT (Scale-Invariant Feature Transform) and SURF (Speeded Up Robust Features) have been largely used for computer vision tasks. Ruiz-del-Solar et al. used SIFT to extract local interest points from both query and reference samples to build a writer-dependent classifier [29]. After extracting interest points from both images, they generated a set of 12 features.

8. **Pseudo-Dynamic Features**

   Oliveira et al., presented a set of pseudo-dynamic features, based on graphometric studies: Distribution of pixels, Progression - that measures the tension in the strokes, providing information about the speed, continuity and uniformity, Slant and Form - measuring the concavities in the signature [30].

2.2.4 Automatic Feature Extractors

There has been an increased interest in recent years on techniques that do not rely on hand-engineered feature extractors. Instead, feature representations are learned from raw data (pixels, in the case of images) using deep learning models.

Early work applying representation learning for the task used private datasets and did not report much success: Ribeiro et al. used RBMs to learn a representation for signatures, but only reported a visual representation of the learned weights, and not the results of using such features to discriminate between

genuine signatures and forgeries [31]. Khalajzadeh used CNNs for Persian signature verification, but only considered random forgeries in their tests [32]. Considering work that targeted the classification between genuine signatures and skilled forgeries, we find two main approaches in recent literature:

1. learning writer-independent features in a subset of users, to be used for training writer-dependent classifiers, or
2. learning feature representations and a writer-independent system at once, using metric learning [33].

Hafemann et al. proposed a Writer-Independent feature learning method [34], where a development set D is used to learn a feature representation φ(X). This representation is learned using a Convolutional Neural Network (CNN) to discriminate among users in D. After the network is trained, the function φ(X) is used as a feature extractor for the exploitation set E, for which Writer-Dependent classifiers are trained. In later work [35], the authors also proposed a multi-task framework, where the CNN is trained with both genuine signatures and skilled forgeries, optimizing to jointly discriminate between users, and discriminate between genuine signatures and forgeries.

Zhang et al. proposed using Generative Adversarial Networks (GANs) for learning the features from a subset of users. In this case, two networks are trained: a generator, that learns to generates signatures, and a discriminator, that learns to discriminate if an image is from a real signature or one that was automatically generated. After training, the authors used the convolutional layers of the discriminator as the features for new signatures. Rantzsch et al. proposed a Writer-Independent approach using metric learning. In this approach, the system learns a distance between signatures. During training, tuples composed of three signatures are fed to the network: (Xr, X+, X−), where Xr is a reference signature, X+ is a genuine signature from the same user, and X− is a forgery (either a random or skilled forgery). The system is trained to minimize the distance between Xr and X+, and maximize the distance between Xr and X−. The central idea is to a learn a

feature representation that will therefore assign small distances when comparing a genuine signature to another (reference) genuine signature, and larger distances when comparing a skilled forgery with a reference.

2.3 SIGNATURE CLASSIFICATION

Different pattern recognition techniques including Support Vector Machines (SVMs), Neural Networks (NNs), Convolutional Neural Networks (CNNs), Hidden Markov Models (HMMs), etc. have been used for signature verification. Distance measuring approaches such as Dynamic Time Warping (DTW) have also been proposed. SVM is a relatively new classifier with better linear classification results than HMM. HMM and DTW have found to be suitable for non-linear classification [36]. NNs are easier to use, whereas Template Matching (TM) and structural techniques have found to be hectic containing immense calculations.

2.3.1 Hidden Markov Models

Several authors have proposed using Hidden Markov Models for the task of signature verification [37][38][39][40]. In particular, HMMs with a left-to-right topology have been mostly studied, as they match the dynamic characteristics of American and European handwriting (with hand movements from left to right). In the work from Justino, Oliveira and Batista, the signatures are divided in a grid format. Each column of the grid is used as an observation of the HMM, and features are extracted from the different cells within each column, and subsequently quantized in a codebook. In the verification phase, a sequence of feature vectors is extracted from the signature and quantized using the codebook. The HMM is then used to calculate the likelihood of the observations given the model. After calculating the likelihood, a simple threshold can be used to discriminate between genuine signatures and forgeries, or the likelihood itself can be used for more complex classification mechanisms.

2.3.2 Support Vector Machines

Support Vector Machines have been extensively used for signature verification, for both writer-dependent and writer independent classification, empirically showing to be the one of the most effective classifiers for the task [41]. In recent years, Guerbai et al. used One-Class SVMs for the task. This type of model attempt to only model one class (in the case of signature verification, only the genuine signatures), which is a desirable property, since for the actual users enrolled in the system we only have the genuine signatures to train the model. However, the low number of genuine signatures present an important challenge for this strategy [42].

2.3.3 Neural Networks and Deep Learning

Neural Networks have been explored for both writer dependent and writer-independent systems. Huang and Yan [43] used Neural Networks to classify between genuine signatures and random and targeted forgeries. They trained multiple networks on features extracted at different resolutions, and another network to make a decision, based on the outputs of these networks. Shekar et al. [44] presented a comparison of neural networks and support vector machines in three datasets. More recently, Soleimani et al. proposed a Deep Multitask Metric Learning (DMML) system for signature verification [45]. In this approach, the system learns to compare two signatures, by learning a distance metric between them. The signatures are processed using a feedforward neural network, where the bottom layers are shared among all users (i.e. the same weights are used), and the last layer is specific to each individual, and specializes for the individual. In the work of Rantzsch et al., a metric learning classifier is learned, jointly learning a feature representation, and a writer independent classifier

Chapter 3

PROBLEM DEFINITION

3.1 PROBLEM CONTEXT

Signature is a behavioral biometric which, unlike physiological biometrics such as iris scan or thumbprint, can be learnt with practice. Therefore, it is much easier to forge a signature. Increasingly more authentication systems are being built around physiological biometrics [46][47][48][49] for this very reason, however, signature still remains the most widely accepted form of personal identification in many legal transactions, such as bank cheques, insurance documents, legal papers, and other everyday paperwork. This widespread use of signatures is due to the fact that, as research has shown [1], people are generally more willing to share their signatures compared to other biometrics.

Being a behavioral biometric, the shape of signature not only depends on current emotional or physical state of the author, but it also evolves with time. This means that there is a significant intra-class variation in genuine signatures of same author.

Another factor which affects shape of signature is the writing material being used, especially in case of offline signatures, because different pens and papers give leave strokes of different sizes, which introduces variability into signatures and can potentially impact the verification accuracy.

3.2 TYPES OF SIGNATURES

Conventionally, there are three different types of forgeries to take into account; random, simple and skilled forgeries. Another category of signatures that have a potential of being misused for fraud are disguised signatures. Based on how

and by who a signature is written, signatures can be divided into following five categories:

3.2.1 Normal Signature

This is the original signature of a person performed by the same author in a regular fashion.

3.2.2 Disguised Signature

This signature is performed by the original author, however the author intentionally does it differently from routine so that they can later claim that it was forged.

3.2.3 Random Forgery

Random forgery is written by a person who does not know the shape of the original signature.

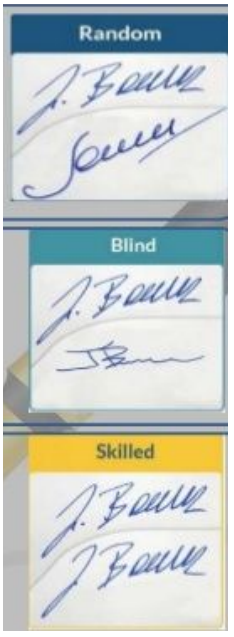


Fig 2: Forgery Types

3.2.4 Simple Forgery

Simple forgery is written by a person who knows the shape of the original signature, but does not have much practice doing it. This type of forgery is also called blind forgery.

3.2.5 Skilled Forgery

Skilled forgery is a suitable imitation of the genuine signature which the writer performs after practicing the original signature.

3.3 CHALLENGES

Now that we have given this problem a context and defined the main terminologies, in this section we discuss the major challenges faced in handwritten signature verification.

3.3.1 Challenges in Signature Extraction

Completely automatic, reliable and accurate extraction of handwritten signature from bank cheques is a challenging problem because of a number of factors.

The complex and varying backgrounds of bank cheques make it extremely challenging to find a generalised thresholding algorithm which can completely separate foreground from background for every kind of bank cheque. Some backgrounds are more easily removed with global thresholding methods, while others with adaptive thresholding methods. Illumination and lighting effects, combined with the cheque backgrounds make this problem even more difficult. Correct thresholding is important because accuracy of all subsequent steps depends directly on performance of this step.

The second major challenge in extraction of signature from the cheque image is to identify which of the foreground pixels belong to the signature and which do not. Foreground elements on the cheque include all the printed text, logos, text lines, and handwritten text including date, amount, name, and signature. Especially challenging here is distinguishing between the signature and other handwritten text, because sometimes a signature looks just like all other handwritten text. No technique exists in literature which can automatically distinguish between signature and other handwritten text, and some kind of priori information, such as knowledge of general area where signature is most likely located, is necessary to extract signature correctly.

Text lines and any other foreground elements which are overlapping with the signature can pose another challenge in signature extraction.

3.3.2 Challenges in Feature Selection

In signature verification, intra-class variations are the first major challenge that any practical classifier should be robust to. The signature of a person varies from time to time and has slight variations which depend on a lot of factors,

including mood, time, tiredness level, etc. No two signatures by the same person are exactly identical, unless one was copied from another sample by some method, which would make it a forgery. Some people's signatures vary more on each try than other people, and these variations themselves are different in nature for different people. For example, for some people the aspect ratio and center of mass of the signature remain almost similar but inclination of signature varies, while for other people aspect ratio and center of mass change too. Features which remain mostly unchanged across different samples of signatures from the same author are called stable features and can be used to get a good classification for that user. However, it is not always necessary that the same set of features which are stable for some authors would be stable for all authors.

Detecting random forgeries is not that much of a challenge, and blind forgeries are comparatively easy to detect too. However, the severity of this problem which arises from intra-class variation is compounded significantly when we throw skilled forgeries in the mix too. Since skilled forgeries are done after considerable practice of the original signature, detecting skilled forgeries while at the same time leaving room for intra-class variations makes this a non-trivial pattern recognition problem.

Disguised signatures make this problem even more challenging, and dealing with disguised signatures is a completely separate domain in signature verification. Disguised signatures, which are written by the original author, should not be classified as forged, however they are specifically designed to look forged. This means that the same set of features which can detect skilled forgeries with high accuracy can fail to adequately detect disguised signatures.

Making signature verification robust to the writing material used is another challenge. Stroke sizes and shapes both change with type of pen and paper used for writing the signature. One method to deal with this is to skeletonize the signature,

that is, thin all strokes to one pixel width. However, some people argue that thinning can lead to loss of other important signature features.

Therefore, designing a feature extractor which is robust to not only intra-class variations but can also distinguish between disguised and forged signatures, while at the same time being invariant to the pen and paper type used is a very significant pattern recognition problem.

3.3.3 Challenges in Signature Classification

Correctness of signature classification depends majorly on the degree of generalization of the features extracted from signature images, however, there are some other factors which need to be considered here.

Firstly, in real world situations, only a few samples of reference signatures for each user are available for training the system. For example, in case of banks, these are the signature samples collected by banks from each person at time of account opening, and are only about 2-4 in number. However, a lot more training samples are required by current state-of-the-art classifiers to give meaningful results. Therefore, some kind of data augmentation is necessary to increase the number of reference signatures, without which performance would be abysmal. Augmentation however can never be as good as real signature samples, and in some cases, it can even have a negative impact depending on how the signatures were augmented.

Secondly, for training only genuine signatures are available in real world scenarios. Much of the research work done around signature verification treats this as a multi-class problem where more than one class of signatures for each author is available when training the classifier. But it is impractical to assume that banks will have forgeries for all of their customers available too. They only have a few genuine signature samples, which makes signature verification a one-class problem in reality. This has a significant negative impact on accuracy of the classifier,

because multi-class classifiers are known to perform generally better than one-class classifiers [50].

Chapter 4

METHODOLOGY

4.1 ITERATIVE DEVELOPMENT APPROACH

4.1.1 Requirements Engineering

Requirements engineering is the process of collecting all the information about what is expected from a software system and documenting and reviewing this information by the stakeholders.

We carried out requirements engineering in four phases as described below:

1. *Feasibility study*: We carried out a study to see whether our proposed system addresses the needs of bank keeping in view the budgetary constraints. We found out that bank professionals (cashiers) are in urgent need of such a system that automatically verifies handwritten signatures.
2. *Requirements elicitation and analysis*: Requirements elicitation is the process of gathering information about the existing similar systems and processes, so that the software to be developed exactly fits the context of the environment in which it needs to be deployed. We studied about existing signature verification systems used in banks abroad such as ParaScript SignatureXpert. Also, we started to see systems proposed in academic papers and their limitations. We analysed these problems and solutions in context of real banks in Pakistan. We asked about the workflow of cashiers and people who do the data entry of account holders in banks. We asked how many signature samples are actually scanned and stored as reference signatures in banks in Pakistan.
3. *Requirements specification*: We then listed the requirements formally and in detail in a document called SRS (Software Requirements specification).

4. *Requirements Validation*: We also checked for the completeness and accuracy of specifications.

4.1.2 Design and Implementation

We developed the system in an iterative manner. So, design and implementation was quite interleaved.

We did following types of design:

1. Architecture Design: We defined a high level architecture in this phase. We decided that client-server architecture best suits our system, if need it to be flexible and scalable.
2. Interface Design: We decided here which modules will interact with each other in what ways.
3. Component Design: We decided here what functionality will each module perform exactly.
4. Database Design: Actually, our database here is a structure of files. So we decided the hierarchy of files used by the system to store its executable code and data.

4.1.3 Testing

As we followed an iterative development approach, so testing was interleaved with the implementation. We constantly looked for bugs and fixed parts of code that could break the system by using exception handling.

We also carried out the testing of our machine learning algorithm, by comparing its classification performance on standard datasets such as GPDS-960 which is the biggest available signature images database. We compared our algorithm's performance with the state of the art accuracy. And our system performance matches the state-of-the-art performance.

4.2 HOW WE CAME UP WITH A SOLUTION

Before starting the project we met domain expert such as Branch manager of a famous local bank. We told about the system and asked about the need for such system. He told that banks are in urgent need of such a system, but accuracy is the key. He told that introduction of such system must not hinder in day to day operations of the bank. And must not increase the task of the cashier or the person scanning the Signature Specimen cards.

Thus, we got the idea of developing an end to end fully functional system that we can deploy with little changes to any bank. And a system that would scale easily as the number of bank account holders increases. The work of cashier is not increased as system only needs a scanned image of the check and MICR code of check. MICR is already read by MICR reader, to check if that bank cheque is genuine.

Our vision was that we should develop such a user friendly system that cashier only has to scan the check and all the heavy lifting is done by the system at the backend. And the cashier gets the results displayed on his computer screen. The work of the person (account holders data entry person) who scans signature specimen cards is not increased, As, he previously had to scan only one card per account holder. Now again, after deployment of our system, he only has to scan one page (having 8 genuine signature samples collected at time of account opening). And the system automatically crops out and saves the genuine samples of a user. In fact the system only keeps one genuine ample image as it only needs the deep feature vectors of all genuine samples.

4.2.1 Initial Experiments

We started off with grid based approaches. But we quickly shifted focus to deep learning approaches as they are the state-of-the-art in image recognition, classification and have been successfully used in many areas such as voice recognition, face verification etc.

We found that for a generalized and scalable system, we must use a Writer independent approach. We researched the literature and learned about the state of the art in image classification and recognition. Thus we found out the deep learning models are found to perform very well in scenarios where there are lots of intra-class variations (just like in our problem, i.e., signatures of one person vary from time to time). We came across [35] in our research, which trained a deep CNN to distinguish between genuine signatures of 531 users of thee GPDS-960 dataset (which is the largest available signature dataset of real users till date, according to our best of knowledge). The the last softmax layer of the trained classifier network was removed. The 2048 activations on the fc7 layer (last layer just before the softmax), were taken as writer independent features. This is a reasonable assumption because these activations were the means by which the network used to decide and classify the signatures. Thus, these were the deep features that captured the unique handwriting characteristics of people.

For all the genuine signature specimens of signatures (taken by the bank at the time of account opening), these 2048 features are computed. This is a very cheap in terms of computation and a near constant time as forward pass of an already trained neural network is constant time and computationally cheap operation. These feature vectors for each specimen of a person is stored in a file on the bank server. When a bank cheque arrives, the check is scanned. The user ID is extracted from MICR (Magnetic Ink Character Recognition) and signature is detected and extracted automatically by the system from the scanned check image. Thus the cashier does not need to perform any extra laborious task. The system actually returns the answer to the question, “Is the signature in the scanned image written by the author whose user ID is this”? Then the image is sent to bank server along with the user ID (Bank account number in this case). Thus, the 2048 deep features are extracted from the questioned signature. A one class SVM is trained on the features of genuine signature samples of that specific ID and the feature vector of questioned signature is predicted. Thus, we get a probability score of weather the signature belongs to the genuine class or not (is a forgery). This

probability as well as a boolean decision (Genuine or Forged), is returned by the bank server to the cashiers computer and is displayed on a GUI. Thus, The cashiers task of signature matching and verification is simplified. He may or may not accept a signature as genuine, which the system has flagged as “Forged”.

The boolean decision returned by the system based on the probability score returned by the One Class SVM is based on a global threshold value set at the time of system development. Its value has been set empirically to a value where False Acceptance Rate equals the False Rejection Rate and is termed as Equal Error Rate. For banks we may want false rejection higher than false acceptance, as a genuine signature flagged as forged may not cause any financial loss as the signer may be requested to sign again.

When the complete system was developed using Sabourin approach. We got interested in doing more research to improve the accuracy further and make ur system as robust as state-of-the-art commercial systems, such as, Parasctipt’s SignatureXpert. So, we found siamese network approach. We also saw that siamese networks have been successfully used in face verification. One example system is Google Facenet. To test our idea, we used the FaceNet trained and optimized on facial images for one shot learning. We gave signature images as input What we found was that the network classified the genuine signatures as more similar to specimen genuine signatures and classified forged signatures as similar to signatures of other User IDs. Thus the FaceNet was able to differentiate between genuine signatures and skilled forgeries.

We then trained the FaceNet on GPDS-960 signature dataset. We again used its deep features and gave them to one class svm just in the previous Sabourin et al. approach we have discussed. What we found was that the results surpass state-of-the-art accuracy.

Now in the next section, we describe, in detail, the algorithm that we have implemented.

4.3 OUR ALGORITHM

The signature verification is a complex problem and it comprises of following subproblems:

- Automatic and exact signature extraction from scanned bank cheque image
- Signature verification i.e. classification of signatures as genuine or forged.

The algorithm which we have developed and used to solve these subproblems is described in complete detail in the following sections.

4.2.1 Cheque Segmentation to Extract Signature

The verification algorithm takes image of a bank cheque as input, which must be segmented to extract the signature from it. To this end, we trained a classifier to distinguish between signature pixels and non-signature pixels.

4.2.1.1 Training and Testing

The Tobacco800 dataset [51][52] contains 1290 images of scanned documents containing machine printed text, logos and handwritten text including signatures. First 50 images from this dataset were manually edited to get two images from each document, one containing only the handwritten text, while the other containing all machine printed text and logos.

Eight-point connected component analysis was the performed on each these images, and SURF features at 400 Hessian threshold were then computed for each of these components.

Multiple classifiers including K-Nearest Neighbor (KNN), Linear SVM, SVM with RBF kernel, a Stochastic Gradient Descent (SGD) classifier, and a

decision tree were trained with these SURF features from signatures and non-signatures.

Experiments were performed on rest of the documents of Tobacco800 dataset and different images of bank cheques to compare these classifiers. KNN and the decision tree gave least false positives and detected almost all of the signatures correctly. However, the size of a KNN model is huge depending on the size of training data, and it takes more time to predict than the decision tree. Therefore, we kept the decision tree for usage in our system.

4.2.1.2 Usage

1. The bottom-right quarter of the cheque image is cropped out because the signature is generally located in this region.
2. Using the Adaptive Gaussian Threshold with block size 25, background from the cheque image is removed to get a binary image.
3. A heuristic which looks at each scan line to see if it contains text lines is used to remove all text lines. Any horizontal or vertical scan line which is made up of 25% or more foreground pixels is removed by marking each pixel as the background pixel. Any pixels on these removed lines whose neighboring pixels are foreground pixels are not marked as background. In this way, the strokes of handwritten text crossing the scan lines are preserved.
4. After removing lines and filling any broken strokes, 8-way connected component analysis is performed for all the remaining foreground pixels.
5. Then, for each of the connected component, SURF features are computed using 400 Hessian threshold. These features are the passes to the trained decision-tree classifier, which marks some of the strokes as signature and the rest as non-signature. All

non-signature pixels are marked as background, leaving us only the signature.

6. A bounding box is computed around the remaining foreground pixels, and the signature is cropped out.

4.2.2 Preprocessing

The extracted signature, which is a binary image, is thinned using the algorithm proposed by Zhang et al. in [53]. This makes the signature invariant to pen and paper type used. Next, the signature is centered inside a 150x220 white canvas such that aspect ratio of the signature is preserved. This makes our algorithm robust to changes in scale, and it becomes invariant to the variations in font size of an author's different signatures.

4.2.3 Feature Extraction

The signature images from the last step are fed to the trained neural network proposed and made publicly available by [35] which has input tensor shape of 150x220. This neural network computes a writer-independent feature vector for the input image, comprised of 2048 values.

Convolutional Neural Networks (CNNs) are the current state-of-the-art in image classification, and using a deep feature extractor instead of handcrafting features ourselves increases verification accuracy greatly and makes the

4.2.4 User Enrollment

To enroll a new user in the system, eight signature images are obtained on a 4x2 grid, and signatures extracted and preprocessed as described above. Then, each of these eight signature image is augmented by slightly skewing the signature image, and randomly rotating it for ±5 degrees, to generate five images. This gives us a total of 40 signature images, for which feature vectors are computed as described above, and stored in a database under the new user's unique identifier.

4.2.4 Signature Verification

For verification, a scanned image of a bank cheque and a user ID is received as input. Signature is extracted from it and preprocessed as described in sections 4.2.1 and 4.2.2 respectively. A 2048-value feature vector representing writer-independent features of the signature is computed from this signature similarly as above. Then, 40 feature vectors of the user whose ID is given are retrieved from the database and used to train a one-class SVM. The trained SVM is then used to classify the test signature extracted from cheque as either forged or genuine. The SVM returns an integer distance value from the hyperplane fitted on the reference feature vectors. If this distance is a negative value, then the signature is forged, otherwise genuine.

4.4 RESEARCH METHODOLOGY

In this project, different approaches to off-line signature verification were investigated and based on experimentation results, best approaches were implemented as a software system.

Experiments conducted aimed to determine the effects of various image preprocessing techniques, absence and presence of image preprocessing, stability of local, global and geometric features, and performance of different models trained with each extracted feature set.

The experiments were conducted with standard datasets obtained from respective research bodies, as well as datasets collected by ourselves.

Accuracy of different SVMs and CNNs, among others, was tested with extracted feature sets with aim of outperforming the state-of-the-art techniques and their results described in [7]. The block diagram below sketches our research methodology.

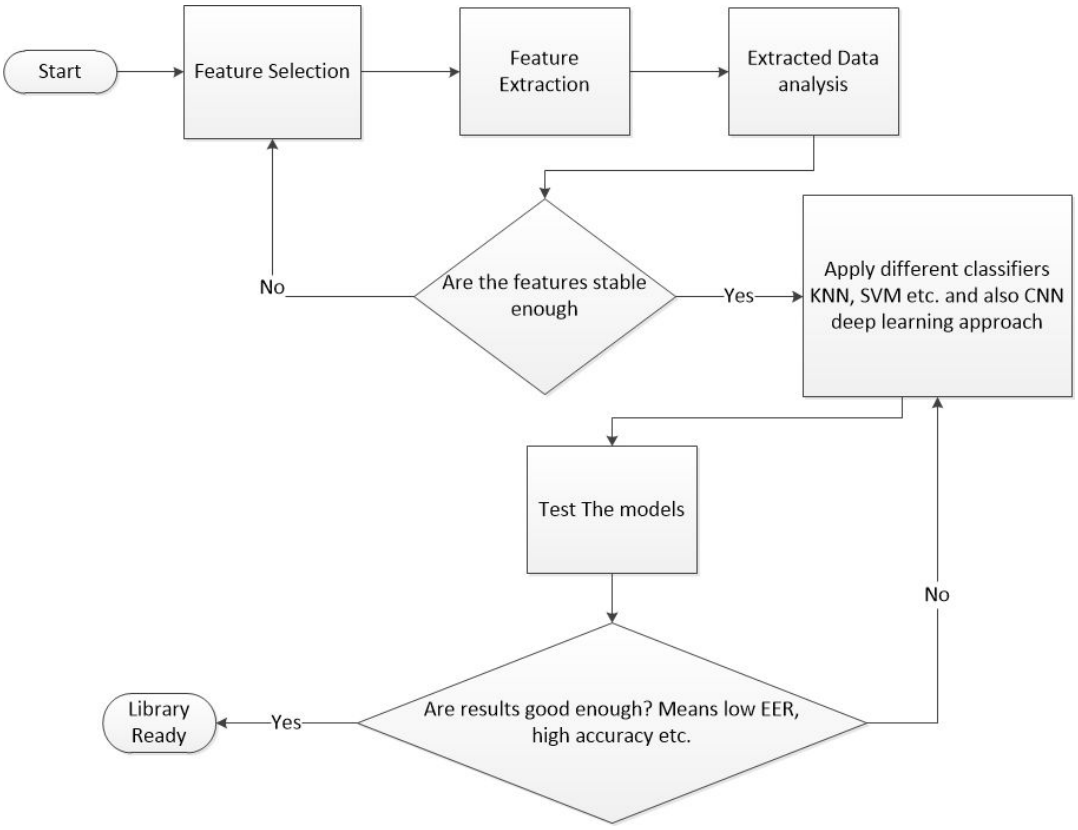


Fig 3: Research Methodology

Algorithms designed was implemented in Python programming language, and packaged as a standalone Python package which was used by our two system front-ends, the desktop application and the Android application.

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

At its highest level, this system has a modular architecture as depicted in the Figure 1 below. The persistent data is stored in a file-based database on a server. The system has two front-end interfaces, a desktop application and an Android application, both of which communicate with a shared Core API for their operations and communication with the database. The desktop application and the Core API reside on the same machine and all calls are made directly. In case of the Android application, an intermediate PHP server is used to help the app communicate with the API over a network.

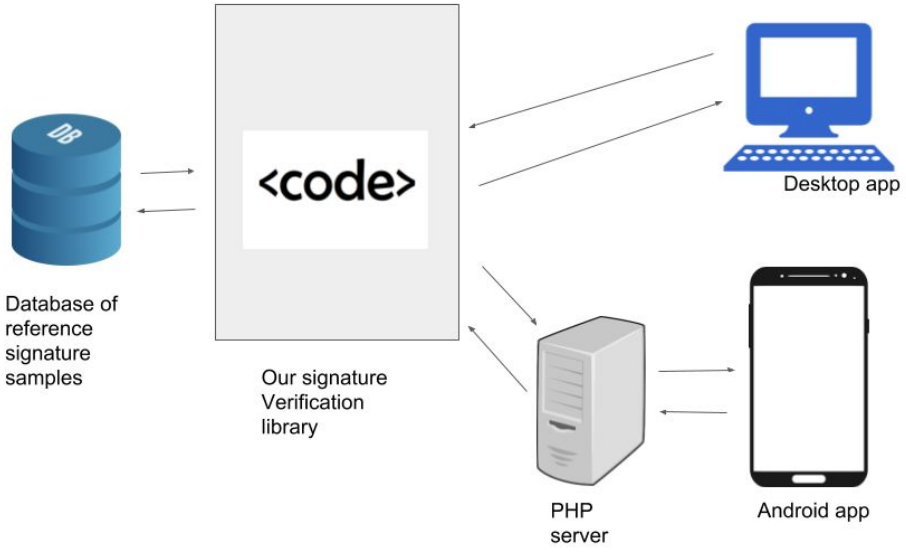


Fig 4: High-Level System Components

5.1.1 Architecture Design Approach

Before starting development we had an open discussion about the system architecture best suited for our system. We started off with thinking about using a

mobile device (android smartphone) and its camera to scan and verify signatures. But this had inherent issues of scalability, reliability, availability, security and robustness. So, we switched to a client server model. All our core libraries and signature verification code is placed on the server. And client only need to send a signature image to our server. The server performs the verification (in case of plane signature image with white background), or extraction+verification (in case of handwritten signature written on a complex cluttered background such as bank cheque). The server also provides enrollment of new users capability. This architecture allows our system to be highly scalable and flexible. We can use all sorts of clients with this system. Only our client needs to capture a signature image (using camera in mobile devices and scanner in desktop clients). Thus we can change the client application without doing a single change to our core library code. This helps in code robustness.

We used a writer independent approach for signature verification, so the bank can enroll as many users as it wants without need to change anything in the system. Thus the system is highly scalable.

Our system is highly flexible, as we can change the signature verification algorithm at the server. And improve our system, without changing anything on the client side applications increasing reliability as there will be few bugs after change.

We wrote our code in the form of modules, so that each module interacts with other module seamlessly. This reduced our system development time and also resulted in a system with high cohesion in the modules and low coupling between them.

5.1.2 Architecture Design

Our system has four main components. First component is login functionality, second component is signature extraction from hank cheques, third is signature verification and the fourth is noise removal and signature preprocessing. The login functionality is introduced in the system so that only authorised users,

such as managers, are able to enroll new people into the system. The cashiers are only able to verify that a given signature belongs to a specific person or not and they cannot enroll new user. The second functionality is signature extraction from the bank cheque. This is important due to the fact that the more accurately we extract the signature from the bank cheques, the more the accuracy of our verification algorithm. The signatures are extracted from the bank cheque, then they are preprocessed by the preprocessing module, before being sent to the verification module. The verification module extracts deep features from the signatures and based on these deep features it gives the decisions of the signature being genuine or false.

5.1.3 Subsystem Architecture

The core API which houses the algorithm central to this system is divided into the modules shown in Figure 2.

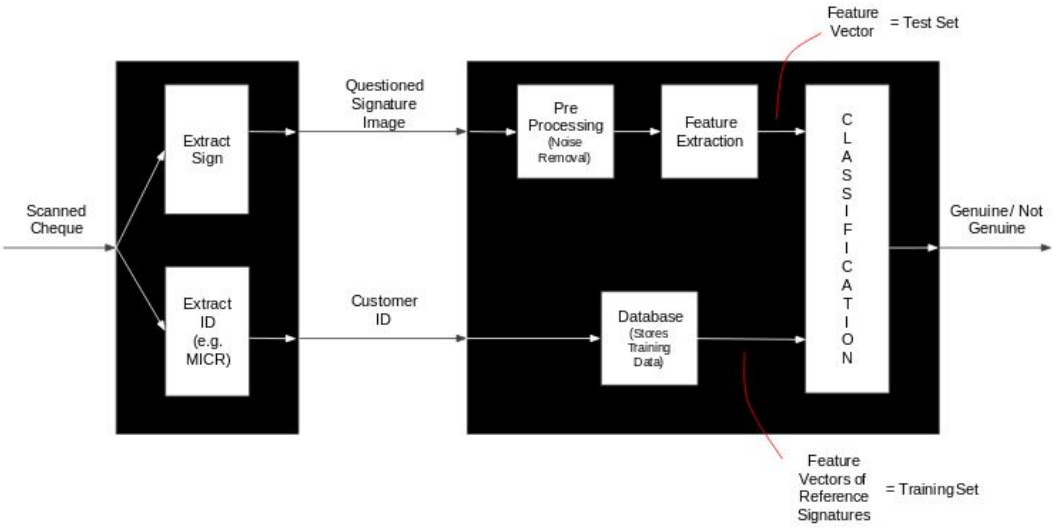


Fig 5: Subsystem Overview

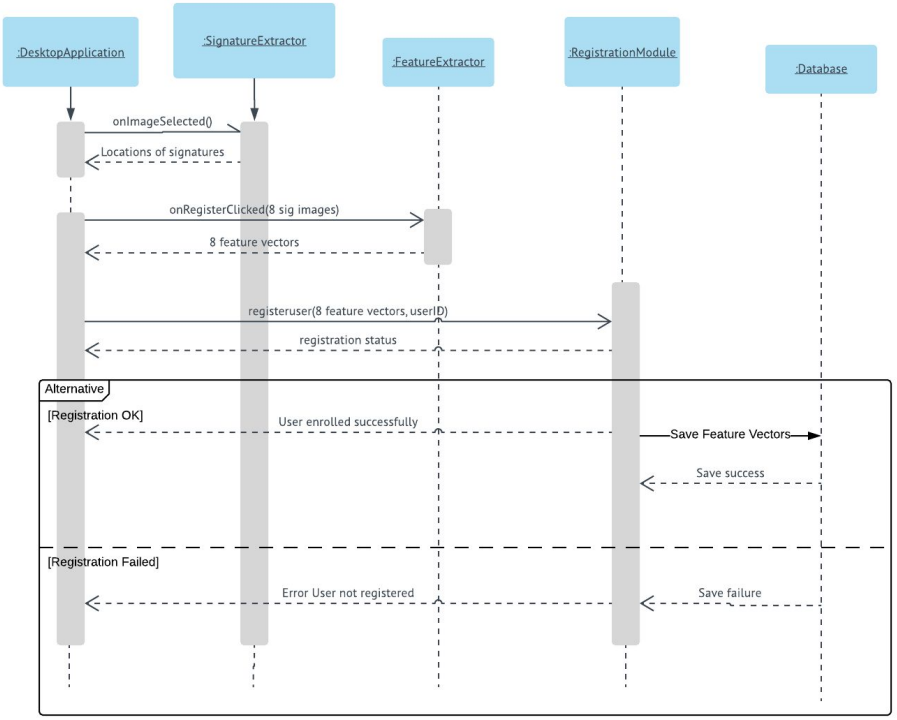
As can be seen above, the system takes a scanned cheque as input, from which signature image is extracted by the extraction module, which along with the user ID is fed to the verification module. Verification module sends the user ID to database to get reference models of the given user, and the signature image is processed and a feature vector is computed. This feature vector, along with the

reference feature vectors are then fed to a classifier which decides whether the new signature matches with the model learned from reference signatures of the user. A binary output is returned by the system.

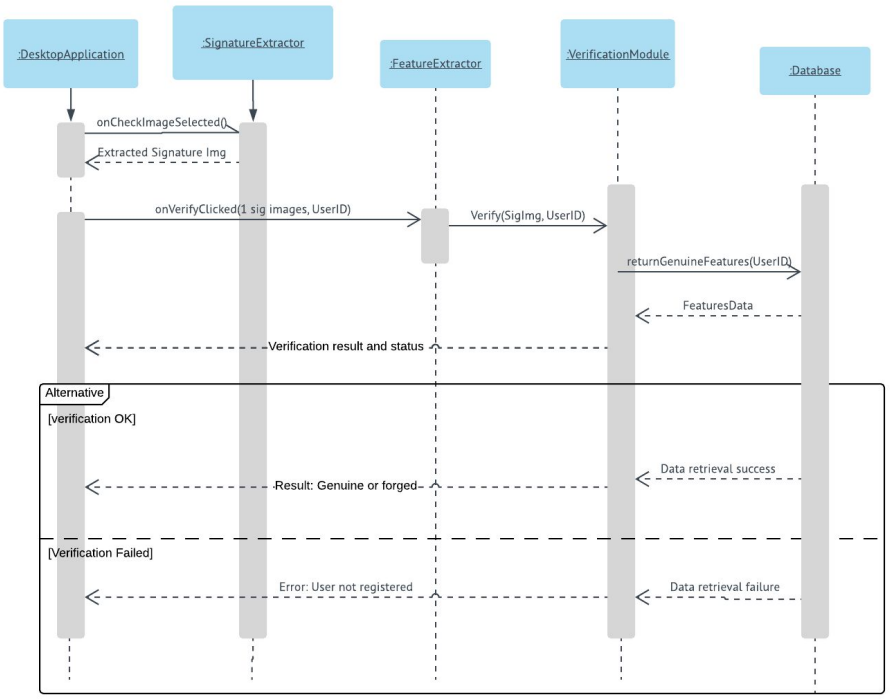
5.2 DETAILED SYSTEM DESIGN

The diagrams below describe the working of the registration and verification modules, showing exactly how these components get their work done.

5.2.1 Registration Sequence

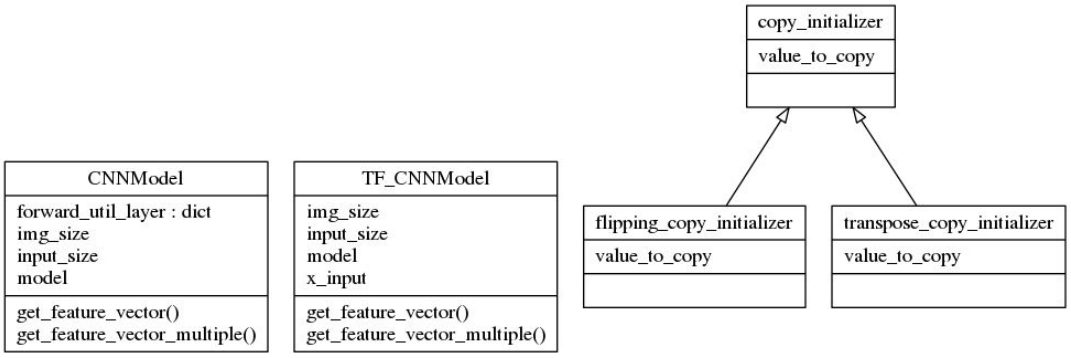


5.2.2 Verification Sequence

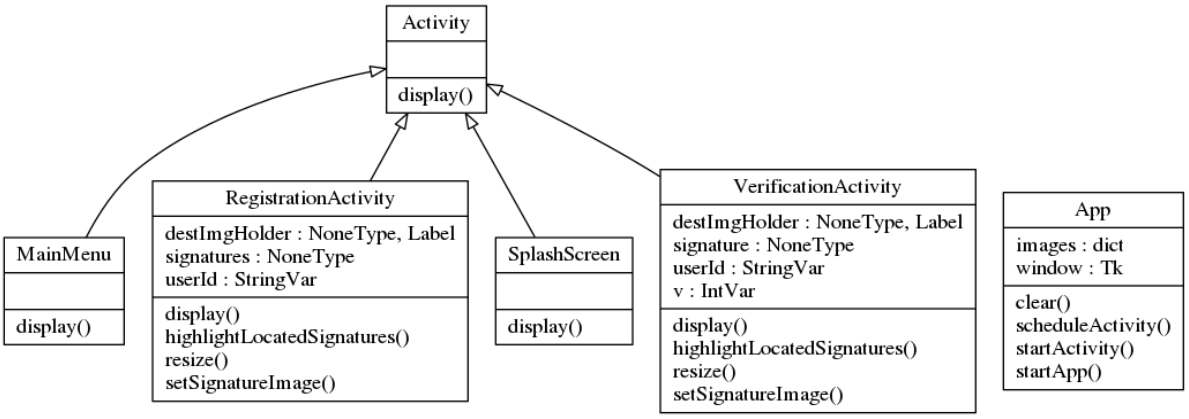


5.2.3 Class Diagrams

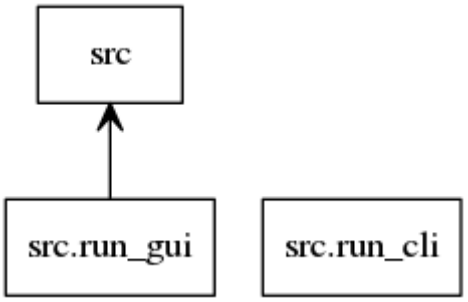
5.2.3.1 CNN Subsystem



5.2.3.2 Desktop Application



5.2.3.3 Package Diagram of Desktop Application



Chapter 6

IMPLEMENTATION AND TESTING

We have developed a complete end-to-end system for offline signature verification for bank cheques, which is comprised of a core verification algorithm which uses machine learning, and two different interfaces for using this algorithm to perform signature verification, in shape of a desktop application and a mobile application.

6.1 TOOLS AND TECHNIQUES

This system was developed in an iterative method, with our team collaborating through GitHub. The following tools and techniques were used during the development of this system:

6.1.1 Programming Languages

The main programming languages used are Python, Java and PHP. The core machine learning algorithm was written in Python language, as well as the desktop application. Java, along with XML was used for development of Android application, which uses a RESTful API to communicate with the server written in PHP.

6.1.2 Core Libraries

Different modern libraries were leveraged to implement core functionalities of the project, including Python Imaging Library (PIL), OpenCV and Scikit-Image for image processing, and TensorFlow and Scikit-Learn for deep learning and other machine learning functions.

6.2 CORE FUNCTIONALITIES

Core functionalities of the project were developed in a modular fashion, which were then packaged in a Python library which is used by both our desktop

application and the Android application for their operation. These functionalities can be divided into three atomic modules:

- A signature extraction module that extracts the signatures from bank cheques. This module receives a scanned image of a bank cheque, removes all background information and all printed and handwritten text excluding the signature, and returns only a binary image of a black signature on white background.
- Feature extraction module takes a binary image of signature, analyzes it using a CNN trained to extract writer-independent features, and returns a feature vector of 2048 numeric values. When enrolling users in the system these features are stored in a database.
- A verification module, which already has the extracted features of genuine signature specimens of system users, extracted and saved at the time of user enrollment. During verification, the system extracts features from the questioned signature. A one-class SVM is trained on the reference features of the user, read in from the database, and then this one-class SVM is used to classify the questioned signature as either genuine (i.e. belonging to referent user) or non-genuine. The one-class SVM returns a probability value, and a threshold value is used to distinguish signatures as being genuine or forged. If this probability is greater than the threshold, the signature is considered genuine, else not.

6.3 OPERATIONAL DETAILS

Next, we discuss the operational details of the developed system, which will narratively demonstrate the core functionalities described above, when used in context of the complete end-to-end system.

This system has two main use cases, enrollment of new users in the system, and verification of signature of an enrolled user. Both are considered separately below.

6.3.1 User Registration

First, we discuss the enrollment module of the system. A new user must be registered with the system using this module, before their signatures can be verified using the verification module.

When a new user is to be added in the system, they are give an A4-sized white sheet of paper with a 4x2 grid printed on it. The new user gives eight samples of their genuine signatures on this sheet, which is then scanned using a scanning device, and fed to our system. In case of a bank, signature acquisition would be done when a customer wants to open a new bank account, and the bank personnel would then scan that sheet of paper and feed it to our system at a later time. A unique identifier for this new user is manually provided to the system along with the scanned image of the specimen sheet. This unique identifier would be the bank account number in the banks' use case.

Upon receiving these two inputs; a unique user identifier and an image of eight signatures on the specified specimen paper, the system works behind the scenes to complete the registration process. If the provided user identifier already exists, an error message is displayed. All eight signatures are automatically located and then extracted from the specimen sheet. Specimen sheet with located signatures outlined is show in the software, to provide feedback to the software user who can use it to verify that the signatures are being located correctly.

The eight signatures are extracted into eight separate images which are preprocessed as discussed in Chapter 4, and then writer-independent features from each of these eight signatures are then extracted, by giving these signatures to the CNN-based feature extractor. This gives us eight 2048-sized feature vectors, called the reference feature vectors, which are saved in a filesystem-based database under the unique identifier of this user. This concludes the user registration sequence.

6.3.2 Signature Verification

In this section, we discuss the signature verification module, which takes either an image of a bank cheque with a signature on it, or an image of a signature only, and an ID of the user, and verifies if the provided signature belongs to the user with the given ID or not.

When an enrolled users' signature comes for test, the system checks if the user with provided ID is registered or not. If the second input is image of a bank cheque, then the signature is automatically located and extracted from this cheque. Once the system has the image of the signature only, this image too is preprocessed like the reference signatures in previous section. Then the same CNN-based feature extractor as above is used to extract 2048 deep features, called the test feature vector, from the questioned signature. The eight reference feature vectors of the specified user are taken from the database and used to train a one-class SVM, which is then asked to compare the test feature vector with the learned model from reference feature vectors classify the questioned signature as either genuine or forged. The verification result is displayed by the system.

6.4 IMPLEMENTATION PLAN

Now, we talk about the detailed implementation plan of a project that was followed by us. We started off with a general overview of the systems functionality. Then we went to a local bank branch and did a need analysis for this system. We interviewed the branch manager and told that we are planning to make an automated handwritten signature verification system that can tell apart genuine from forged signatures. He raised a concern that the genuine signature samples of people vary over time so how come it's possible that the system can detect the natural variations that occur in the handwriting of people over time. We said that there are approaches in machine learning that help capture variations in data and can detect anomalies and a forged signature is essentially a forgery. He said that banks are in desperate need of such system and if we are successful in developing such a system he will definitely recommend the system to his bank. We then broke

down the system into components and developed it component by component first of all we focused on the verification module that is the heart of the system. For verification we first started off with classic approaches of machine learning such as grid based approach. The grid based approach was not robust enough to capture the variations that occur in the signatures. So, we switched towards the modern techniques of deep learning. We first learned the techniques of deep learning and then tried of a number of techniques from literature. In literature, a deep CNN was used for signature verification problem. The author of the system trained a CNN on a signature dataset named GPDS-960. He took 531 users randomly from the dataset and train the CNN as a classifier. So that, job of the CNN was to distinguish between the signatures of 531 users. The author's hypothesis was that if a convolutional neural network is able to distinguish between the users, it means that it has captured certain deep features that characterize the signature. He he trained the CNN into formulations first was a classifier only using the genuine signatures. In the second formulation, he trained the neural network in a way that the signature image, the signer ID, and a label i.e. genuine or forged, was given to the network. For the label, there was additional output that was binary. 1 told it was forged, and zero told it was genuine. This way the network learn to classify the signatures as genuine or forged. And also was able to distinguish between different users. So, the features learnt by the system essentially was something that can tell about genuine signatures from forgeries to test is hypothesis it took the FC 7 layer feature vector from the train Network and trained a one-class SVM on it. Then predicted the signatures of a validation set using this approach. This approach surpassed state-of-the-art. We mainly used this approach, but we tried other networks also. We did an experiment on Google facenet. Google facenet was designed for one shot learning of facial images. We took the pre-trained model, that was trained on the facial images. We tried it on signatures. What we saw was, that the network recognised the genuine signatures as similar to original specimen signatures and forged signatures as dissimilar. That was really the problem we were trying to solve, so we decided to train the face that on GPDS-960 dataset.

Retrain the Resnet model on the GPDS-960 dataset of Signature images. Then we used the resnet deep features (fc7 layer) that characterize a signature and trained the SVM in a similar way as described above. The results show that this approach surpasses, the state-of-the-art accuracy, in signature verification on standard datasets.

Now, we come towards the signature extraction from bank cheque. For this we used an approach that is based on SURF (Speeded Up Robust Features) features. For this the check image is converted to grayscale and then is binarized. The connected components of the binary image are extracted for each connected component. The SURF features are computed. Feature vectors of each connected component is predicted by a nearest neighbour classifier that compares the feature vectors with a database of SURF features descriptors of handwritten connected components and connected components in a machine printed image. So, if most of the feature vectors of a connected component are closer to a machine printed text features then that connected component is white out. So, only handwritten text remains on the bank cheque image. The handwritten text contains the courtesy amount and other information as well as the handwritten signature. We are only interested in the handwritten signature in this case so we use the prior knowledge of bank cheque format. We crop out the right bottom of the image so we get the signature image. The signature image is reconstructed using morphological operations so that the parts that are lost during connected component classification stage are reconstructed. Then the signature image is fed to the verification module for verification. The result of verification is then displayed on the GUI (Graphical User Interface).

Chapter 7

EXPERIMENTS AND RESULTS

To design the signature algorithm, we conducted a number of experiments which are discussed in this section. However, before going into detail of experiments, we shall define some terminologies.

Accuracy of a classifier is defined as the ratio of correct decisions made by it to total decisions it was asked to make. This, however, is not a very effective measure of performance.

False acceptance rate (FAR) is the ratio of number of forged signatures accepted as genuine to the total number of forged signatures. **False rejection rate (FRR)** is the ratio of number of genuine signatures rejected as forged to total number of genuine signatures.

Classifiers usually return a probability value, and a decision is taken based on an adjustable threshold value. Changing this threshold changes the false acceptance and false rejection rates. The point at which false acceptance rate becomes same as the false rejection rate is known as the equilibrium point, and the error value here is known as the **equal error rate (ERR)**.

In [35], Sabourin et al. proposed a CNN-based feature extractor which returns a 2048-value writer-independent feature vector. This feature extractor, which was trained on signatures of 531 users from GPDS 960 dataset, was made publicly available by its owners. In their research, they used Linear SVM and SVM with RBF kernel to classify signatures based on features extracted with this network. However, these classifiers require multiple classes at time of training, i.e. both genuine and forged signatures, which is not practical in real situations.

Therefore, we decided to test Sabourin's network with a one-class SVM with RBF kernel.

Sabourin has proposed two different models, namely signet and signet_f, both of which are publicly available, along with the features extracted with these models from GPDS, MYCT, and CEDAR datasets. We took these features extracted with signet model and gave them to the one-class SVM. Following results were observed

Dataset	Accuracy
GPDS	93.13%
MYCT	84.70%
CEDAR	93.64%

Table 1: Accuracy of One-Class SVM on Signet Features

The datasets above contain only skilled forgeries and no disguised signatures. To test the performance of Sabourin's model with disguised signatures, we used the signature datasets from ICFHR (See Table 2). The signatures in these datasets were preprocessed as described in our methodology section, and then features were extracted from each of these datasets using both Sabourin's signet and signet_f models. Equal error rates observed are summarised in Table 2.

DATASET	FACENET	SIGNET	SIGNET_F (lambda=0.95)	SIGNET_F (lambda=0.999)
SigComp2009	12.981	13.622	14.103	-
SigComp2010	70.556	70.556	71.111	70.556
SigComp2011/Chinese	37.602	23.433	23.706	23.706
SigComp2011/Dutch	17.214	17.214	17.214	16.119
SigComp2012	15.385	15.385	19.414	14.286

Table 2: Equal Error Rates on ICFHR datasets

Facenet [54], which is a network based on ResNet architecture [55], was originally developed for facial recognition. During our experiments, we tried using the Facenet model trained on faces to classify signatures, and observed that it was giving promising results. Therefore, we retrained Facenet on signatures. The facenet was trained on genuine signatures of first 531 users from GPDS database. It is supposed to be used as a classifier, however, instead of doing that, we took embeddings calculated by facenet from images as 128-value feature vectors and used these to classify signatures with our one-class SVM. As can be seen in Table 2 above, this gave results comparable to Sabourin's models, and sometimes even better.

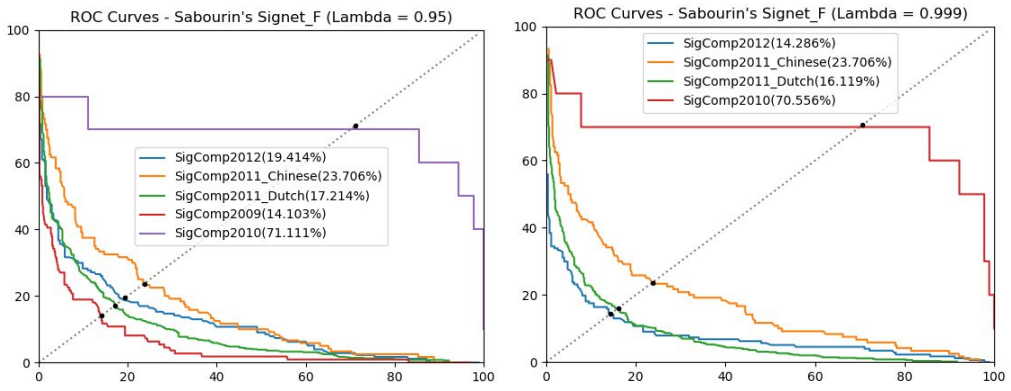


Fig 6(a): ROC curves for signet_f model

The figures above and below show ROC curves of ICFHR datasets when features were extracted from them with Sabourin's signet, signet_f with lambda 0.95, then with lambda 0.999, and finally with our own facenet.

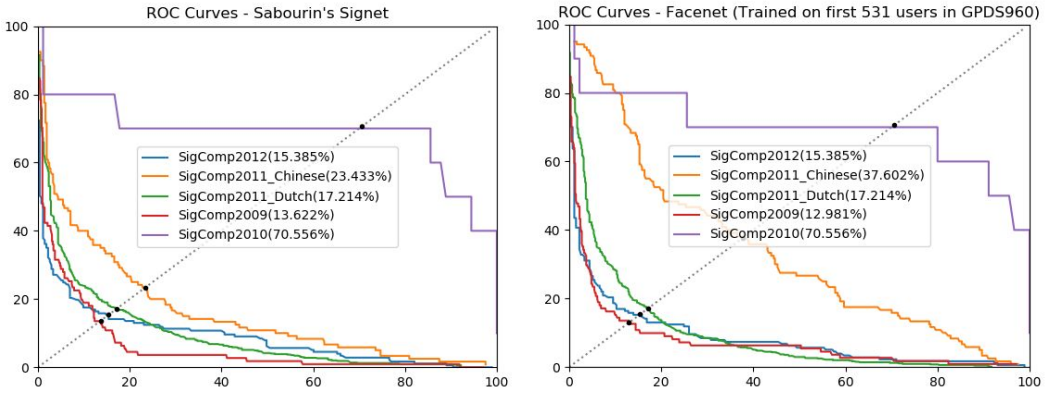


Fig 6(b): ROC curves for signet and facenet models

Chapter 8

CONCLUSION AND FUTURE WORK

The problem statement that has been described in the start of this report has been addressed. We have developed a complete end-to-end solution that consists of user enrollment module, signature extraction, signature verification. So, this end to end system can be easily deployed in a bank; with little or no changes to the existing system. Just the dependencies need to be installed in the bank's current system. There is no need for the bank to deploy a server with GPU, as we will use a trained model.

Predicting from a trained model does not require high compute and it is a near constant time operation. The system that we have proposed is a writer independent approach. So, no matter how many people open account in a bank everyday, the system does not need to be retrained.

Our system easily scalable to an indefinite number of users without change, also the signature images that were given at the time of account opening can also be deleted to save the space and only one image per user can be used just for the human verifier to compare with, if he sees a suspicious signature. The system instead of only giving hard decisions genuine or forged, gives the probability also, so that in case the system detects a forged signature, the human verifier can see the probability. And based on that probability he can decide to accept or reject the signature and hence the bank cheque. This system can reduce the financial losses that occur every year due to forged signatures. This system is not intended to replace the human verifier, but it is to assist the human verifier, so that, human error can be reduced. Human errors occurs due to many factors such as mood, tiredness etc. So, this system can predict the signatures with a uniform accuracy and so chances of accepting a forged signature is greatly reduced.

We can work on the system in the future to enhance the performance of the signature Extraction module as well as the signature verification module. We can enhance the performance of signature extraction module by using the flilformity approach by Dijezi et al. This approach is inspired by human visual system, as to how it can effortlessly distinguish between handwritten and printed text. We can also work for better reconstruction of signatures after removal of the printed line (line printed on cheques marking the place of signatures).

For future work one can enhance the signature verification accuracy by doing more and more research. But the core system does not need to be changed, because the system is flexible enough to incorporate all the changes easily without any change in the pipeline. We have described an experiment to use Google FaceNet's embeddings (feature vectors for one-shot learning of facial images). Thus, we can refine this approach and train the FaceNet Siamese Network on GPDS-960 offline handwritten signature images. We can also do transfer learning on already trained Facenet and then retrain it on signature images. This may result in a better performance.

Chapter 9

REFERENCES

[1] Jain, A., Hong, L., & Pankanti, S. (2000). Biometric identification. *Communications of the ACM*, *43*(2), 90-98.

[2] Kalera, M. K., Srihari, S., & Xu, A. (2004). Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence*, *18*(07), 1339-1360.

[3] Al-Omari, Y. M., Abdullah, S. N. H. S., & Omar, K. (2011, June). State-of-the-art in offline signature verification system. In *Pattern Analysis and Intelligent Robotics (ICPAIR), 2011 International Conference on* (Vol. 1, pp. 59-64). IEEE.

[4] Check Fraud Statistics & Techniques You Should Know About. (2017, July 07). Retrieved from http://www.relyco.com/blog/laser-check-printing/check-fraud-statistics-techniques-know/

[5] Malik, M. I. (2015). Automatic Signature Verification: Bridging the Gap between Existing Pattern Recognition Methods and Forensic Science.

[6] Radhika, K. R., Venkatesha, M. K., & Sekhar, G. N. (2010). Off-line signature authentication based on moment invariants using support vector machine. *Journal of Computer Science*, *6*(3), 305.

[7] Impedovo, D., & Pirlo, G. (2008). Automatic signature verification: The state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(5), 609-635.

[8] Malik, M. I., Liwicki, M., & Dengel, A. (2011, September). Evaluation of Local and Global Features for Offline Signature Verification. In *AFHA* (pp. 26-30).

[9] Madasu, V. K., Mohd. Hafizuddin Mohd. Yusof, Hanmandlu, M., & Kubik, K. (2003, December). Automatic Extraction of Signatures from Bank Cheques and Other Documents. In *DICTA* (Vol. 3, pp. 591-600).

[10] Djeziri, S., Nouboud, F., & Plamondon, R. (1998). Extraction of signatures from check background based on a filiformity criterion. *IEEE Transactions on Image Processing*, *7*(10), 1425-1438.

[11] Zhu, G., Zheng, Y., Doermann, D., & Jaeger, S. (2007, June). Multi-scale structural saliency for signature detection. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (pp. 1-8). IEEE.

[12] Duchon, C. E. (1979). Lanczos filtering in one and two dimensions. *Journal of applied meteorology*, *18*(8), 1016-1022.

[13] Canny, J. (1987). A computational approach to edge detection. In *Readings in Computer Vision* (pp. 184-203).

[14] Mandal, R., Roy, P. P., & Pal, U. (2011, September). Signature segmentation from machine printed documents using conditional random field. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (pp. 1170-1174). IEEE.

[15] Jiang, R., Al-Maadeed, S., Bouridane, A., Crookes, D., & Beghdadi, A. (Eds.). (2016). *Biometric Security and Privacy: Opportunities & Challenges in the Big Data Era*. Springer.

[16] Novaković, J. (2016). Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav Journal of Operations Research*, *21*(1).

[17] Goshtasby, A. (1985). Description and discrimination of planar shapes using shape matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 738-743.

[18] Khotanzad, A., & Hong, Y. H. (1990). Invariant image recognition by Zernike moments. *IEEE Transactions on pattern analysis and machine intelligence*, *12*(5), 489-497.

[19] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.

[20] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). Ieee.

[21] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.

[22] Ojala, T., Pietikainen, M., & Harwood, D. (1994, October). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on* (Vol. 1, pp. 582-585). IEEE.

[23] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2548-2555). IEEE.

[24] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012, June). Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (pp. 510-517). Ieee.

[25] Pavelec, D., Justino, E., Batista, L. V., & Oliveira, L. S. (2008, March). Author identification using writer-dependent and writer-independent strategies. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 414-418). ACM.

[26] Pal, S., Blumenstein, M., & Pal, U. (2011, February). Off-line signature verification systems: a survey. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology* (pp. 652-657). ACM.

[27] Satyarthi, D., Maravi, Y. P. S., Sharma, P., & Gupta, R. K. (2013). Comparative study of offline signature verification techniques. *International Journal of Advancements in Research & Technology*, *2*(2), 1-6.

[28] Sabourin, R., & Genest, G. (1994, October). An extended-shadow-code based approach for off-line signature verification. i. evaluation of the bar mask definition. In *Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on* (Vol. 2, pp. 450-453). IEEE.

[29] Ruiz-del-Solar, J., Devia, C., Loncomilla, P., & Concha, F. (2008, September). Offline signature verification using local interest points and descriptors. In *Iberoamerican Congress on Pattern Recognition* (pp. 22-29). Springer, Berlin, Heidelberg.

[30] Oliveira, L. S., Justino, E., Freitas, C., & Sabourin, R. (2005, June). The graphology applied to signature verification. In *12th Conference of the International Graphonomics Society* (pp. 286-290).

[31] Ribeiro, B., Gonçalves, I., Santos, S., & Kovacec, A. (2011, November). Deep learning networks for off-line handwritten signature recognition. In *Iberoamerican Congress on Pattern Recognition* (pp. 523-532). Springer, Berlin, Heidelberg.

[32] Khalajzadeh, H., Mansouri, M., & Teshnehlab, M. (2012). Persian signature verification using convolutional neural networks. I*nternational Journal of Engineering Research and Technology, 1*.

[33] Bertolini, D., Oliveira, L. S., Justino, E., & Sabourin, R. (2010). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition*, *43*(1), 387-396.

[34] Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2016, July). Writer-independent feature learning for offline signature verification using deep convolutional neural networks. In *Neural Networks (IJCNN), 2016 International Joint Conference on* (pp. 2576-2583). IEEE.

[35] Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, *70*, 163-176.

[36] Justino, E. J., Bortolozzi, F., & Sabourin, R. (2005). A comparison of SVM and HMM classifiers in the off-line signature verification. *Pattern recognition letters*, *26*(9), 1377-1385.

[37] Dolfing, J. G. A., Aarts, E. H., & Van Oosterhout, J. J. G. M. (1998, August). On-line signature verification with Hidden Markov Models. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on* (Vol. 2, pp. 1309-1312). IEEE.

[38] Coetzer, J., Herbst, B. M., & du Preez, J. A. (2004). Offline signature verification using the discrete radon transform and a hidden Markov model. *EURASIP Journal on applied signal processing*, *2004*, 559-571.

[39] Kashi, R. S., Hu, J., Nelson, W. L., & Turin, W. (1997, August). On-line handwritten signature verification using hidden Markov model features. In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International conference on* (Vol. 1, pp. 253-257). IEEE.

[40] Yang, L., Widjaja, B. K., & Prasad, R. (1995). Application of hidden Markov models for signature verification. *Pattern recognition*, *28*(2), 161-170.

[41] Kumar, R., & Singhal, P. (2017). Review on Offline Signature Verification by SVM.

[42] Guerbai, Y., Chibani, Y., & Hadjadji, B. (2015). The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognition*, *48*(1), 103-113.

[43] Huang, K., & Yan, H. (1997). Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recognition*, *30*(1), 9-17.

[44] Shekar, B. H., Bharathi, R. K., Kittler, J., Vizilter, Y. V., & Mestestskiy, L. (2015, May). Grid structured morphological pattern spectrum for off-line

signature verification. In *Biometrics (ICB), 2015 International Conference on* (pp. 430-435). IEEE.

[45] Soleimani, A., Araabi, B. N., & Fouladi, K. (2016). Deep multitask metric learning for offline signature verification. *Pattern Recognition Letters*, *80*, 84-90.

[46] Jain, A. K., Bolle, R., & Pankanti, S. (Eds.). (2006). *Biometrics: personal identification in networked society* (Vol. 479). Springer Science & Business Media.

[47] Masek, L. (2003). Recognition of human iris patterns for biometric identification.

[48] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM computing surveys (CSUR)*, *35*(4), 399-458.

[49] Ratha, N., & Bolle, R. (Eds.). (2003). *Automatic fingerprint recognition systems*. Springer Science & Business Media.

[50] Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of machine learning research*, *5*(Jan), 101-141.

[51] Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., & Heard, J. (2006, August). Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 665-666). ACM.

[52] Agam, G., Argamon, S., Frieder, O., Grossman, D., & Lewis, D. (2006). The complex document image processing (CDIP) test collection project. *Illinois Institute of Technology*.

[53] Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, *27*(3), 236-239.

[54] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

[55] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI* (Vol. 4, p. 12).