Text Independent Offline Hand Writer Recognition Using Machine Learning

Faraz Ahmad Khan

Department of Computer and Information Sciences

University of Northumbria at Newcastle

This dissertation is submitted for the degree of

Doctor of Philosophy

2017

I would like to dedicate this thesis to my loving family: Masood Ahmed Khan, Dr. Salma Masood Khan, Shireen Fawad, Nayab Faraz, Faizan Ahmad Khan and Zaurayz Ahmad Khan. But my special gratitude goes towards my dear mother and sister, who have always supported me throughout my hardships. You have always lifted me up with your love and moral support whenever I was feeling down and lost. I am forever indebted to you and no words are enough to describe how much I love and value you.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

Faraz Ahmad Khan
2017

Acknowledgements

If the completion of my doctoral research and thesis is a forest, I am only a tree and, as the popular proverb goes, a tree does not make a forest. I want to use this opportunity to acknowledge the numerous trees that contributed to the growth of this doctoral forest.

First and foremost, my deepest gratitude and thanks go to Allah for enabling me to reach this stage and for helping me throughout the many hardships I faced during my stay in UK. Without His generosity and mercy none of this would have been possible and I would not be the man I am today.

I would also like to extend my gratitude to my supervisors: Dr. Ahmed Bouridane, Dr. Fouad Khelifi and Dr. Muhammad Atif Tahir, who have provided me with valuable support, guidance and encouragement. Through their guidance I was able to achieve many things which I would not have been capable of doing alone. I would also like to thank all my colleagues in Lab F7 with whom I have shared countless memories, which I will always cherish.

I would also like to sincerely acknowledge the scholarship and financial support provided by Northumbria University which allowed me to complete this research.

Abstract

Handwriting is a behavioural biometric that an individual learns and develops over time and automated writer identification systems can be developed by identifying these behavioural aspects of an individual's writing style. These writer recognition systems greatly assist forensic experts by facilitating them with semi-automated tools that segment the text, narrow down the search, help with visualization and finally assist in the final identification of an unknown handwritten sample.

Handwriting, as a behavioural characteristic, has been a subject of interest for researchers for many decades and intensive research performed in this field has resulted in the development of multiple methods and algorithms. However, automated writer identification is still a challenging problem. Difficulties in segmenting text and the deviation of an individual from his or her unique writing style is the reason for ongoing research in this field.

This thesis aims to investigate the problems faced in automated writer identification and propose novel techniques of segmentation and classification that would contribute to the field of writer identification. This has led to four different contributions.

First a novel segmentation algorithm is proposed for segmenting sub-words within hand written Arabic words, the proposed method outperforms the previously used projection profile method. The second proposed method offers a segmentation free multi-scale Local Ternary Pattern Histogram for text independent writer identification. Local ternary patterns are applied at various scales to produce a predictor model for its respective scale while the high dimensionality problem of a multi-scale approach has been tackled with dimensionality reduction using SR-KDA. The third contribution tackles the problem of writer identification in noisy conditions. A robust offline text independent writer identification system is proposed using Bagged Discrete Cosine Transforms. The proposed system effectively utilizes discrete cosine transform for writer identification while avoiding problems of high dimensionality and memory limitations. Finally, in the fourth contribution a dissimilarity Gaussian mixture model is proposed for describing the contrast between different writers of a dataset. Furthermore, a weighted histogram approach is also proposed that penalizes bad prediction scores with a cost function to significantly enhance the identification rate.

List of Publications

Journal Papers

- Faraz Ahmad Khan, Muhammad Atif Tahir, Fouad Khelifi, Ahmed Bouridane, Resheed Almotaeryi - "Robust Off-line Text Independent Writer Identification Using Bagged Discrete Cosine Transform Features", Expert Systems with Applications, 2016.
- Faraz Ahmad Khan, Fouad Khelifi, Muhammad Atif Tahir, Ahmed Bouridane - "Dissimilarity Gaussian Mixture Models for Efficient Offline Handwritten Text-Independent Identification using SIFT and RootSIFT Descriptors", IEEE Transactions on Information Forensics & Security, 2018.

Conference Papers

- Faraz Ahmad Khan, Ahmed Bouridane, Fouad Khelifi, Resheed Almotaeryi, Sumaya Almaadeed - "Efficient segmentation of sub-words within handwritten Arabic words", Control, Decision and Information Technologies (CoDIT), 2014.
- Faraz Ahmad Khan, Muhammad Atif Tahir, Fouad Khelifi, Ahmed Bouridane - "Offline Text Independent Writer Identification using Ensemble of Multi-Scale Local Ternary Pattern Histograms", European Workshop on Visual Information Processing (EUVIP), 2016.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

BDCT	Bagged Discrete Cosine Transform
CC	Connected Component
DCT	Discrete Cosine Transform
DGMM	Dissimilarity Gaussian Mixture Model
DoG	Difference of Gaussians
FRG	Feature Relation Graph
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HoG	Histogram of Oriented Gradients
KDA	Kernerl Discriminant Analysis
LBP	Local Binary Pattern
LCS	Longest Common Subsequence
LDA	Linear Discriminant Analysis
LPQ	Local Phase Quantization
LTP	Local Ternary Pattern
MLTPH	Multi-scale Local Ternary Pattern Histogram
PAW	Part of Arabic Word

SGMM	Similarity Gaussian Mixture Model
SIFT	Scale Invariant Feature Transform
SOM	Self Organizing Map
SR-KDA	Kernel Discriminant Analysis using Spectral Regression
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Background

The term 'biometrics' originates from the Greek words; 'bios' and 'metron' meaning life and measurement respectively. Therefore, in a broad sense biometrics can be seen as a measurement of a human's body characteristics (Riera *et al.*, 2009). According to this basic definition, biometrics has been applied in many fields such as pharmacy, biology, engineering, agriculture and medicine to study biological behaviours with the help of statistical methods. However, by the second half of the 20th century and with the development of advanced technological methods, biometrics was given a new meaning. In today's research, the term biometrics refers to a method that automatically analyses behavioural and physiological characteristics of a human in order to recognize that person or distinguish that person from another (Gracia, 2006).

Over the last few decades, biometrics has also acquired a new meaning where rather than focusing on the technique of measurement, the characteristic to be measured is focused on. Hence, biometrics is also defined as a unique and measurable characteristic of a human being that would help to automatically identify an individual.

The main task of any identity management system is to determine or verify an individual's claimed identity. Such a system is necessary to prevent imposters from accessing resources that are not intended for them and thereby rightfully protected. Traditional methods of establishing or verifying an individuals identity involves knowledge based systems such as passwords or token based systems such as ID cards, however the problem with such traditional forms of ID is that they can be easily manipulated, lost or stolen. Such risks of compromising an identity management system do not exist with biometric authentication systems. Biometric recognition systems can actually establish the identity of an individual based on who he or she is rather than what he or she remembers or possesses. Biometric systems also offer the additional advantage of negative recognition, something that is not offered by password and token systems. Negative recognition is the process of establishing whether an individual is already enrolled in the system even though the individual might deny this. Negative recognition is of great importance in welfare or benefits disbursement systems where an individual might try to claim multiple benefits under different identities.

Biometric systems rely on a variety of behavioural and physiological characteristics or traits to establish the identity of an individual. Some systems supplement behavioural traits with physiological traits to produce an extra level of security. Both of these traits are defined briefly below.

Physiological Characteristics

These refer to the physical traits of an individual defined as the measurements of the physical parameters of a specific part of a human body. Physiological characteristics can be captured from finger prints, palm prints, knuckle prints, retina scanning, facial features etc.

Behavioural Characteristics

Behavioural characteristics are related to the behaviour of a human being or in other words how an individual uses his or her body. Behavioural characteristics, although not as strong as physiological characteristics, do provide some advantages. They are non obtrusive and can therefore be collected without the knowledge or consent of the subject. Furthermore, they usually do not require any special equipment of hardware and are therefore comparatively more cost effective (Yampolskiy & Govindaraju, 2008).

1.2 Biometric Characteristics

A characteristic can be physical or behavioural, something that is measurable, and through which the identity of a person can be known or verified. It is essential that these characteristics are intrinsic in nature and may involve what a person does or is. Jain et al. Jain *et al.* (2006), recommended seven factors by which the suitability of a biometric trait can be established.

Measurability

It should be possible to collect and digitize a biometric trait and it's collection should not cause any undue inconvenience to the individual. Furthermore, the data acquired should be processable such that features can be extracted from it.

Permanence

For the characteristic being measured to be reliable, it must be quantifiable and relatively static. Traits being measured must be stable enough so that passage of time, environmental conditions and age do not bring about significant changes in them.

Universality

Every member of the relevant population must be able to provide the biometric being measured and it should also be in a form that can be conveniently collected.

Uniqueness

In biometrics recognition, it is essential that when a characteristic is selected through which an individual may be identified, it must be completely unique. It must have no equal and therefore be the only one in existence. Selection of such a characteristic is key to successfully determine and/or verify the identity of an individual (Jain *et al.*, 2004).

Performance

The resources required to achieve the target recognition accuracy must be within the constraints set by the system. Furthermore, the system should be capable of responding to queries promptly with acceptable accuracy (Ashbourn, 2014).

Acceptability

The population that will utilize the biometric system must accept the system and be willing to present their biometric traits to that system. If the users are not comfortable with the device or sensor being used, they will avoid using it.

Circumvention

It should not be fairly easy to find a “way around” the system. The biometric traits collected should not be easily imitated or mimicked.

All of the above mentioned criteria constitute the characteristics of a very strong biometric. However, it is accepted that no single biometric modality will meet every single one of the criteria mentioned above, but for a biometric system to be acceptable and practical, it should at least be appropriate for its intended application (Impedovo & Pirlo, 2008; Jain *et al.*, 2006).

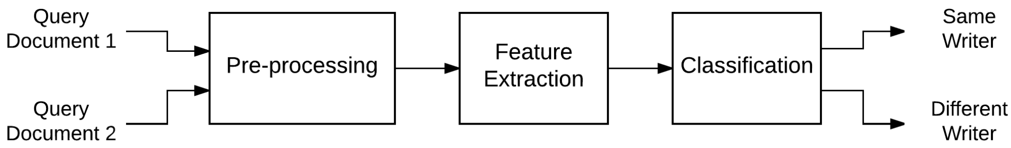


Figure 1.1: Verification mode of biometric systems.

1.3 Modes of Biometric Recognition

A biometric recognition system works in the same way as a pattern recognition counterpart i.e. an individual's biometric data is first collected, features are extracted from the acquired data finally these features are compared against a known database. Depending on the application, biometric systems may operate in verification/authentication or identification mode.

Verification or Authentication mode

Authentication comes from the Greek word 'authentes' which means author. Biometric verification systems aim to prove the identity claimed by an individual. In this mode, the system compares the captured biometric data of a person with his or her own given biometric template already available on the system. Verification comparisons are carried out on a one-to-one basis, where the system compares the given biometric data to a specific template on the system in order to verify the claimed identity. In this case, the system answers the question, "Does the provided biometric data match that of subject X"? This mode is mostly used in positive recognition systems where the objective is to prevent people from using the same identity. A diagram of a verification process is shown in Figure 1.1.

Identification mode

Identification comes from the Latin words 'idem-facere' which means 'to make the same'. In this mode, the system tries to determine the identity of an individual without any previous claim about their identity. This mode performs a one-to-many comparison as the captured biometric data is compared with that of all the available subjects in the database. Therefore, identification biometric systems try to answer the question, "To whom does the provided biometric data belong to"? The identification mode of biometric systems is crucial in negative

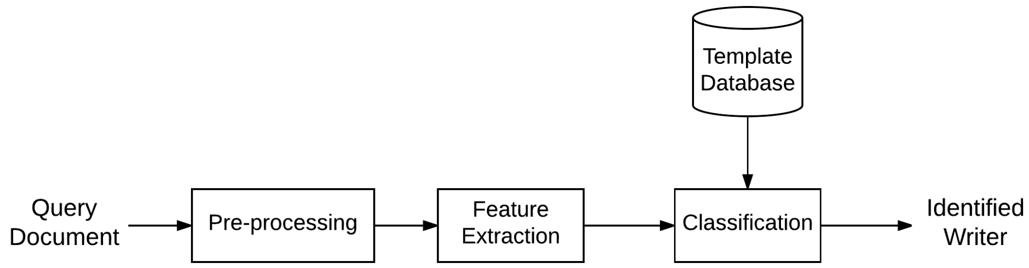


Figure 1.2: Identification mode of biometric systems.

recognition systems, where the aim is to prevent a single individual from using multiple identities. The identification process is shown in Figure 1.2.

1.4 Handwriting as a Biometric Trait

Handwriting is a behavioural biometric that a human learns and develops over time. It involves coordination between the hands (motor effectors controlled by the brain) and the eyes (feedback receptors to the brain). This coordination allows humans to generate complex ink shapes and sequences. Handwriting biometrics is the science of identifying this behavioural aspect of an individual's writing style and using it to develop writer recognition systems. Handwriting, as a behavioural characteristic, has been a subject of interest for researchers for many decades. This interest is spread over a variety of fields that have a collected interest in the handwriting of an individual. Examples include forensic experts, psychologists, palaeographers as some of the few. Writer identification has also been used in domains such as forensic analysis (Tapiador *et al.*, 2004), analysis of historical documents (Fornés *et al.*, 2008; Fornes *et al.*, 2009) and security (Ballard *et al.*, 2006). Furthermore hand writer identification is also used in fields such as verifying the authenticity of financial documents, wills including in criminal investigations where a piece of handwriting is the only piece of evidence available to the police, such as in case of ransom notes. Hand writer recognition algorithms aim to make the task of the forensic experts much easier by facilitating them with semi-automated tools that segment the text, narrow down the search, help with visualization and finally assist in the final identification of an unknown handwritten sample. These techniques are expected to produce a list of predicted authors of the unknown text sample, ranked in terms of confidence measure from which the forensic expert will make the final decision (Fiel & Sablatnig, 2012). This supports the forensic experts as they are no longer required to manually go through every image in huge databases but rather they can make their decision from a small list of writers that are predicted by the algorithm.

The process of identifying an individual based on his or her handwriting may be as old as handwriting itself. Handwriting can be used as a reliable identification tool is of great value to the law and justice enforcement agencies. Along with other forms of evidence, such as DNA, fingerprints and material analysis, handwriting is also recognized as a valid and admissible form of evidence in the court of law for Questioned Document Examiners (QDE) or Forensic Document Examiners (FDE) (Davis, 2007; Jain *et al.*, 2002; Srihari *et al.*, 2002).

Handwriting is considered to be a useful and valid biometric as it has shown to meet the criteria outlined in Section 1.2. It is universal, as in terms of practicality it is considered that a population is capable of handwriting and have also previously produced written documents. It is unique to every individual as demonstrated by (Srihari *et al.*, 2002) and (Arora *et al.*, 2002). Furthermore, it is also stable throughout the life of an individual. (Walton, 1997) observed that although ageing and the physical conditions associated with ageing do have some impact on the handwriting process, they are not enough to prevent or affect the process of identification.

1.4.1 Text-dependent vs Text-independent Identification

The challenge of automatic hand writer identification can be divided into either a text dependent or a text independent approach (Namboodiri & Gupta, 2006). In text dependent approaches the identity of the writer is determined using a specific transcript and usually the writer is asked to reproduce a sample of text. In such systems the availability of the writer is assumed. One example of text dependent systems is that of signature identification systems (Jain *et al.*, 2002).

Text independent systems, on the other hand, aim to identify a writer regardless of the written text. Generally, text dependent systems are more accurate but since they assume the exact same text to be reproduced for accurate identification along with the availability of the writer, they are not always practical. Text independent systems, on the other hand, are more practical as they do not depend on the exact same content but on the other hand do require a large amount of data from every writer in order to properly train the system.

1.4.2 Online vs Offline Writer Identification

Online writer identification involves automatic processing of the text while it is being written using a stylus and tablet or a digitizer pen. This allows the system to have access to a set of additional dynamic features such as the slant of the pen, it's speed and acceleration, pressure on the tip etc. (Li & Tan, 2009; Yang *et al.*, 2016).

On the other hand, offline writer identification does not have the luxury of working with such an abundance of information but only relies on the scanned static images available to it. Consequently, a lot of research in offline writer identification has been focused around identification and extraction of new features from the scanned ink traces available. Online writer identification has a very high accuracy rate due to the additional information available to it and is generally considered much superior to it's offline counterpart systems.

1.5 Hand Writer Identification Process

A writer identification process can be divided into four stages: image acquisition, pre-processing, feature extraction and classification as shown in Figure 1.3.

Image Acquisition

The main goal of this step is to digitize the handwritten documents. For the identification to be effective, it is important that the images acquired at this stage are of an acceptable quality as pre-processing and feature extraction do not work well or even at times fail to work on low quality and noisy images. The tools usually used for this step are either a digital camera or a scanner. The images are usually scanned in at more than 300 dpi as images lower than 300 dpi are considered to be of low quality.

Pre-processing

The majority of handwritten identification algorithms require an effective pre-processing of the acquired documents. The pre-processing stage performs many corrections (if needed) to the acquired images, such as skew correction, space normalization, smoothing, thinning, binarization, padding, region of interest selection, words or connected component extraction and so on. These pre-processing tasks help to improve the final result of the identification system.

Feature Extraction

In its raw image format, there is usually far too much information available for the purpose of writer identification and most of that information is not even relevant and therefore useless for

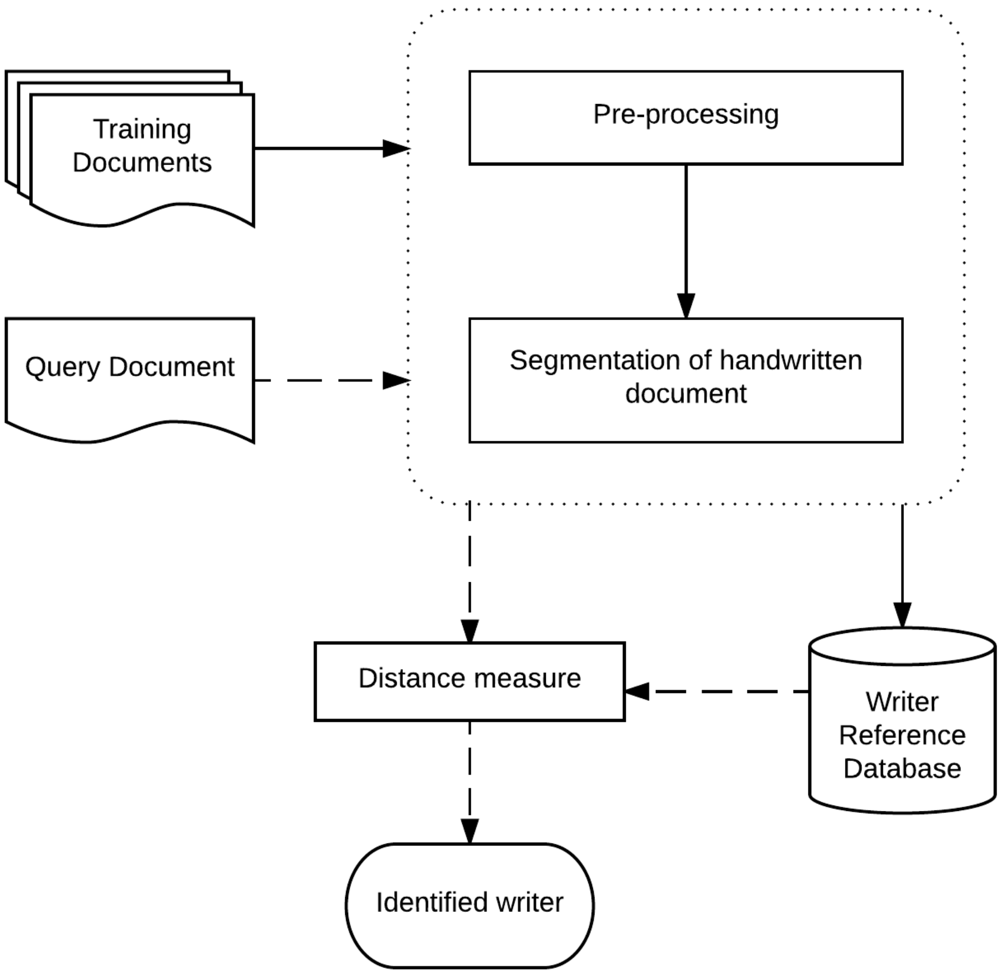


Figure 1.3: Training and testing phases of a typical writer identification process.

the task at hand. This stage therefore, involves determining and then extracting features from the images which serve as useful identifiers for the classification stage. These identifiers are known as features and effective features are those that are able to identify the class to which it belongs with a high degree of confidence.

Classification

In the classification stage, the extracted set of features are analysed in order to determine which class they belong to. In the case of writer identification, each class is an individual writer having its own unique handwriting with a unique pattern of features. This process of attributing an unknown set of features to a class is known as classification. In terms of classification, the features that are extracted to represent a sample are known as it's feature space, and the measurement of these features with respect to other samples are known as feature vectors. Therefore, the distance between feature vectors represent their similarity. Feature vectors belonging to the same class would be close together with small distance, whereas different class feature vectors would be far apart with large distance.

1.6 Problem Statement

Writer identification research is based on the hypothesis that every individual has a unique writing style that is consistent throughout their life and also that their unique writing style is distinguishable from the writing style of another individual. This hypothesis was verified by (Arora *et al.*, 2002) where the authors obtained handwriting samples from 1500 individuals of varying gender, race and age. Global feature extraction was used along with machine learning to verify this hypothesis. However, it has also been known that at times a person will write in a style that deviates from his or her unique writing style. This is attributed to a number of reasons such as the type of pen used to write the text and also the condition in which the writer is writing. Figure 1.4, which is a sample from the CVL database (Kleber *et al.*, 2013) shows the effect of change of pen on an individuals writing style. It can be seen that the writer changed their pen mid-document and with it the writing style also got affected. Similarly, Figure 1.5 (another writer from the CVL database) shows the effect of change of condition during writing such as text written in a hurry or if the writer seated or standing. It can clearly be seen in Figure 1.5 that the writing style of the writer changed mid-document, it appears that the changed text was written in a rushed condition. These problems are the reason for continued research in this field and also the main challenges faced by researchers (Fiel &

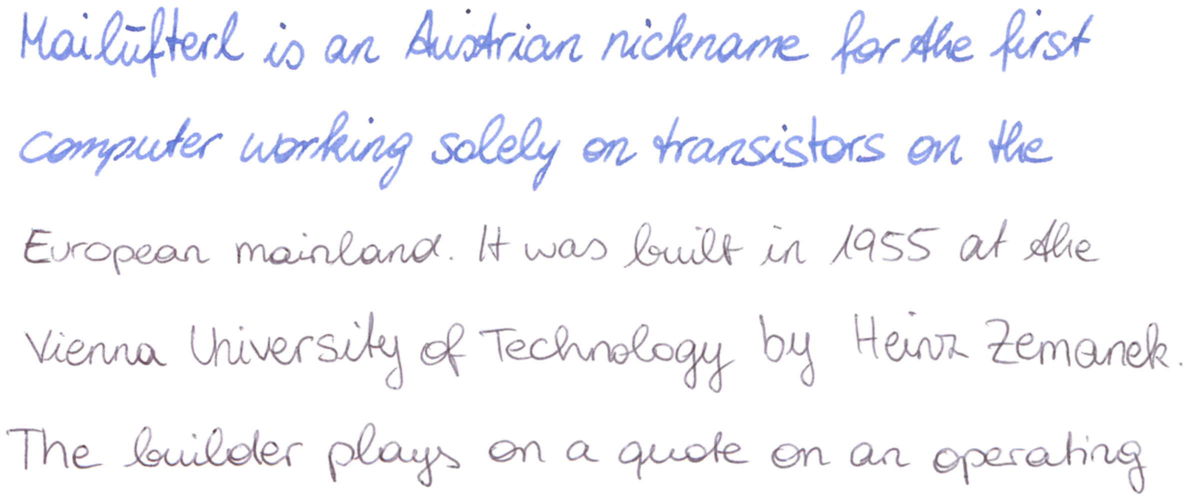


Figure 1.4: Sample from the CVL database showing effect of pen change on writing style.

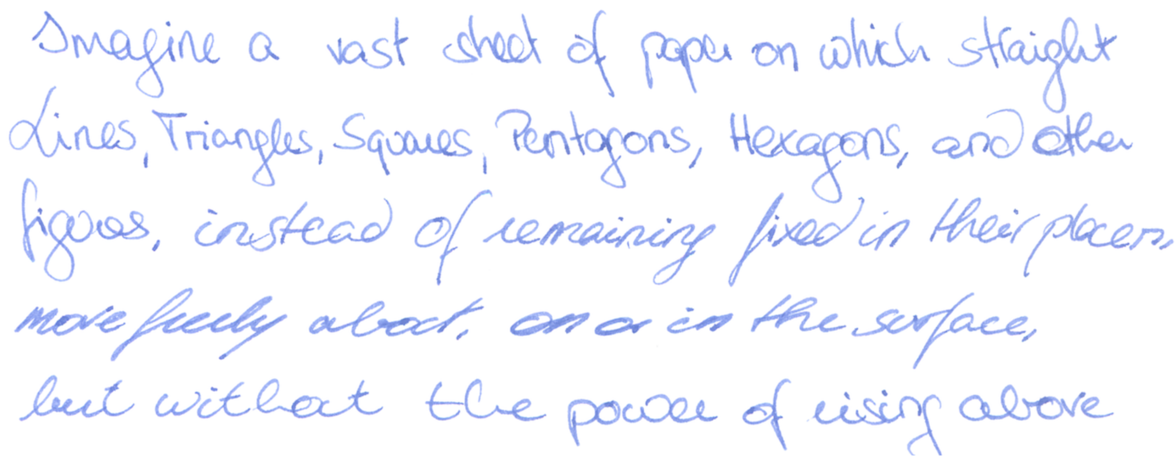


Figure 1.5: Sample from CVL database showing effect of writing conditions on writing style.

Sablatnig, 2013).

1.7 Research Aims and Objectives

This thesis aims to investigate novel techniques of feature extraction and classification in the field of writer identification. The main objective of this thesis is to investigate and further develop techniques of segmentation and classification that would contribute to the field of writer identification. The aims and objectives of this research can be summarized as as follows:

- To develop a new segmentation algorithm that would help to effectively segment sub-words in hand written Arabic text.

- To investigate the concept of multi-scale feature extraction at a textural/global level for use in writer identification.
- To develop a robust writer identification system that can be used by forensic analysts to analyse noisy and distorted documents.
- To propose a writer identification system that effectively combines different features for improved identification performance.
- To investigate a dissimilarity model framework for writer identification. The dissimilarity framework should further improve identification results by producing a strong contrast between correct and incorrect classifications.

1.8 Thesis Organization

This thesis consists of 7 chapters in total, the organization of which is shown in Figure 1.6. The contents of every chapter are summarized below:

Chapter 2 discusses previously published systems in writer identification. An emphasis on offline systems has been made although some notable online identification systems have also been reviewed. The works are organized based on the level of feature extraction performed for their respective systems.

Chapter 3 identifies the problem of segmentation of characters within Arabic words and proposes a novel method of segmenting sub-words in hand written Arabic text. The preprocessing step of binarization is exploited to generate clusters for every character within a word. These clusters are then automatically extracted using the central limit theorem.

Chapter 4. reports a segmentation free multi-scale local ternary pattern histogram (MLTPH) for offline text independent writer identification. Local Ternary Patterns (LTP) have been applied at various scales to produce a predictor model for its respective scale whereas, high dimensionality due to the multi-scale approach has been tackled by using SR-KDA for dimensionality reduction.

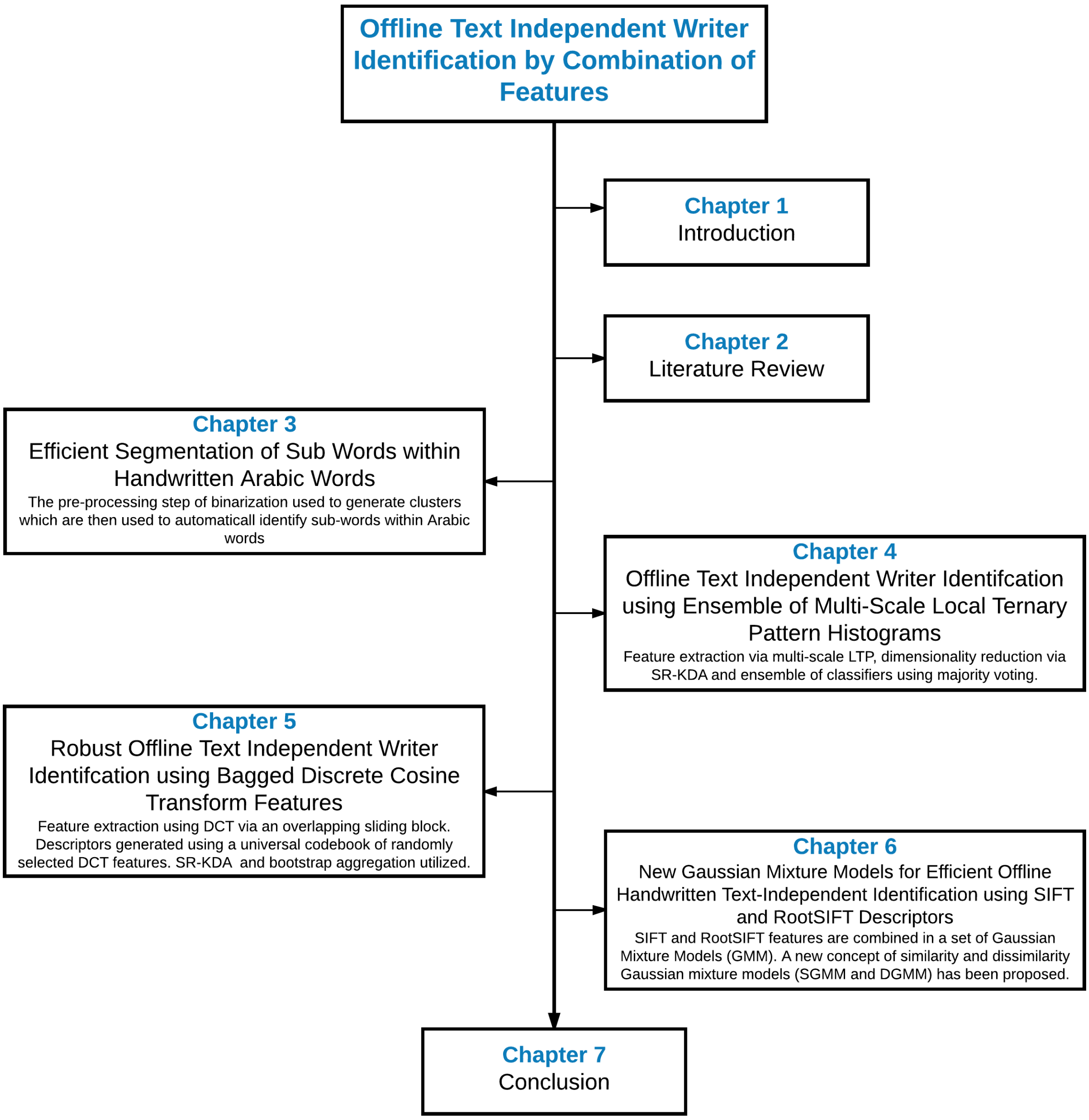


Figure 1.6: Thesis Organization.

Chapter 5 proposes a robust offline text independent writer identification system that can be used by forensic analysts in noisy conditions. A Bagged Discrete Cosine Transform (BDCT) approach has been proposed so that DCT can be utilized effectively for the purpose of writer identification without having to face problems such as memory limitations and dimensionality problems.

Chapter 6 presents an efficient handwriting identification system that combines SIFT and RootSIFT descriptors in a set of Gaussian mixture models (GMM). In particular, a new concept of similarity and dissimilarity Gaussian mixture models (SGMM and DGMM) is introduced. While a SGMM is constructed for every writer to describe the intra-class similarity that is exhibited between the handwritten texts of the same writer, the DGMM represents the contrast that exists between the writer's style on one hand and other different handwriting styles on the other hand. Furthermore, a new weighted histogram method is proposed to derive an intermediate prediction score for each writer's GMM.

Chapter 7 summarizes the main contributions made and also details the conclusions drawn from the work done in this thesis. Recommendations for possible future work are also presented.

Chapter 2

Literature Review

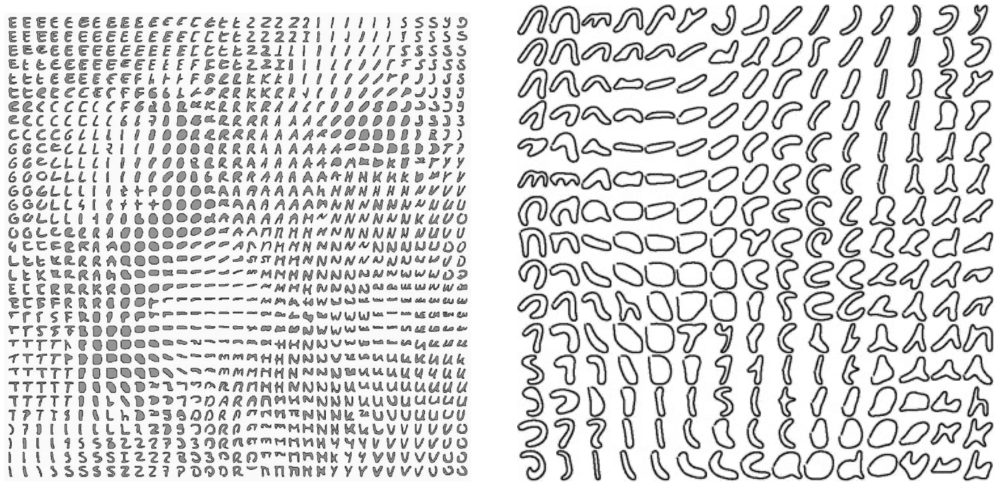
2.1 Background

The identification of subjects based on their handwriting is an active area of research. The challenges posed by the difficult problems encountered in this field have lead to many the advancements in pattern recognition and computer vision. Over the last two decades, considerable advancements have been made in the field of writer identification in a variety of languages. Along with multi-script systems, a lot of work has gone into developing language specific writer identification as well, since each language brings with it its own unique challenges depending on the characteristics of its writing style.

A detailed survey in the field of writer identification can be found in the works done by (Chen, 2012) and (Sreeraj & Idicula, 2011). In the following sections, we perform a brief survey focusing mainly on offline hand writer identification approaches. The survey is organized based on the text component used for extracting writer discriminatory features i.e. characters/graphemes, words, lines or paragraph/page level identification.

2.1.1 Identification based on Graphemes

Rather than focusing on an entire script containing alpha-numeric characters, (Leedham & Chachra, 2003) proposed a writer identification algorithm by considering only on the ten handwritten digits. Features such as height, width, area, centre of gravity, junctions, endpoints and degree of roundness were extracted from the handwritten digits. Hamming distance was used as the distance measure to identify the writers. They achieved an accuracy of 95% on a database of 15 writers who contributed random strings of digits from 0 to 9.



(a) (b)

Figure 2.1: (a) A *33 x 33* self-organizing map of CO3 graphemes (Schomaker & Bulacu, 2004). (b) A *15 x 15* self-organizing map of FCO3 graphemes (Schomaker *et al.*, 2007).

(Schomaker & Bulacu, 2004) proposed a new feature in upper-case western handwriting called Connected-Component Contours (CO3). This feature was used to construct a universal codebook using the Kohonen self-organizing map (SOM). By using the codebook, a descriptor can be computed for each text image based on the occurrence histogram of its corresponding CO3. In order to enhance the identification performance, the authors also combined CO3 with another edge-based feature describing the angle of edges in a histogram. A variant of CO3, called Fragmented Connected Component Contours (FCO3) was also proposed by Schomaker et al. in (Schomaker *et al.*, 2004) and (Schomaker *et al.*, 2007). Investigating codebook generation for writer identification, (Bulacu & Schomaker, 2005) have shown that the K-means clustering technique can also be used for generating the code-book as the performance offered was very close to the one obtained with the Kohonen self-organising map. Recently, FCO3 have also been adopted by (Khalifa *et al.*, 2015) with a multiple codebook approach where the codebook for every writer was divided into 12 sub-codebooks. These multiple codebooks were then used to represent every writer. It was demonstrated that using multiple codebooks to represent every writer produced better results than by using a single codebook approach.

(Van Der Maaten & Postma, 2005) utilized a statistical approach and a model based approach for improving automatic hand writer identification. They argued that statistical approaches have the limitation of only relying on single scale features whereas the model based approach require an extended amount of time to generate writer models. The authors proposed

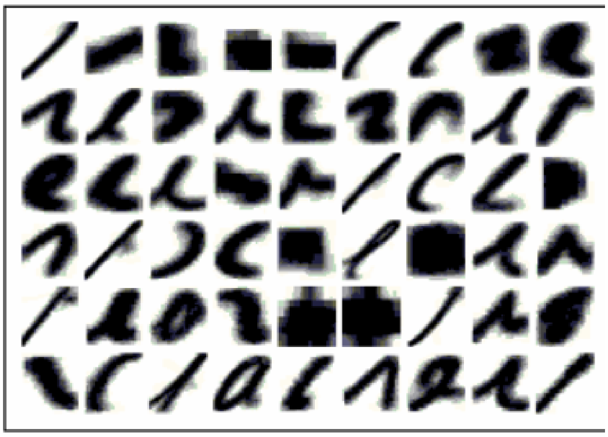


Figure 2.2: Graphemes clustered into writer invariant codebook (Bensefia *et al.*, 2005a,b).

to remove both these limitations. In the case of statistical approach multi scale features were evaluated and from them, the best performing single scale feature was selected. For model based approaches the authors showed that the generation of Kohonen maps was unnecessary and that randomly generated codebooks tend to give better results. They applied their proposed system on the Firemaker database using 100 writers for the codebook generation and 150 writers for testing. They achieved a Top 1 accuracy of 97% by combining their improved statistical and model based approach.

Bensefia et al. (Bensefia *et al.*, 2005a,b) proposed a writer identification and verification system based on extracting strokes of characters of the Latin script and generating a writer invariant codebook. These strokes of connected components were called graphemes. A universal feature environment was generated by clustering the extracted graphemes using k-means clustering. The authors applied their proposed system on three text independent databases, covering 39 historical documents and 88 books and the evaluation considered both writer identification and verification modes. In writer verification, the evaluations were made by comparing the distributions of graphemes and making decisions based on the mutual information between the two distributions, while writer identification was done in an information retrieval capacity. Their proposed method showed an accuracy of 86% when tested on the IAM database using 150 writers. This method of grapheme feature extraction and clustering was also employed by (Schlapbach *et al.*, 2005).

(Gazzah & Essoukri Ben Amara, 2006) performed writer identification by extracting the features at two levels: spacial and frequency domain. Wavelet transforms and entropy were extracted at the frequency domain and features such as height of text, intra and inter word distance, ascender slant etc. were extracted at the spatial domain. Classification was performed by using a Multi Layer Perceptions (MLP) classifier. Their proposed system was tested on a database of 180 text samples achieving an accuracy of 94.73%. Gazzah et al. further improved this result in (Gazzah & Amara, 2007) by using a 2D discrete wavelet transform lifting scheme for feature extraction. An MLP classifier was used achieving an accuracy of 95.68% using the same database of 180 text samples. They also compared their result with a Support Vector Machine (SVM) classifier and concluded that MLP outperforms an SVM classifier.

Writer identification at the character level in Persian script was performed by Baghshah et al. (Baghshah *et al.*, 2006). The handwritten text, was pre-processed and then segmented into strokes where each stroke was further described by a set of features; direction, horizontal and vertical slant and measurement ratio. Fuzzy Learning Vector Quantization (FLVQ) was used for classification. Their proposed system achieved an accuracy of 95% when applied on an online database of 128 writers.

(Bulacu & Schomaker, 2007) identified the writers based on two sets of features. The first set of features were extracted at the texture level and a probability distribution function (PDF) was used to represent the features such as the slant, roundness of the writing style and the curvature. The second set of features operated at the character level and focused on extracting the information at the allograph level where the writers were characterized by a stochastic pattern generator of graphemes. The graphemes extracted from each writer are his/her characteristics. The PDF's of these graphemes were computed using a codebook obtained by grapheme clustering. A combination of these features achieved attractive results as their proposed system showed an accuracy of 89% using the full IAM database of 650 writers. (Siddiqi & Vincent, 2007) also used the codebook concept and improved it by extracting the graphemes at a much smaller scale i.e. at a sub-grapheme level. The authors achieved this result by using a modified form of component by component extraction for the purpose of the codebook generation. A fixed window of size $n \times n$ (13×13 achieved best results) was moved over the text from left to right while keeping the vertical origin fixed. Due to the small scale of the grapheme extraction the authors mentioned that to get an effective accuracy rate each writer would require a large amount of training data to familiarize the identifier. Their proposed system achieved an identification rate of 94% using 50 writers from the IAM database.

Balacu et al. further applied their proposed method mentioned above on the Arabic script as well in (Bulacu *et al.*, 2007). Here allograph level features were also extracted after segmenting the Arabic text into characters. The minima of the lower contour was used for segmentation purposes. They applied their proposed system on the IFN/ENIT Arabic database using 350 writers and achieved a Top 1 accuracy of 88%. They observed and concluded that the identification of text in Arabic script is more difficult than that of Latin scripts.

(Abdi *et al.*, 2010) proposed a writer identification algorithm for the Arabic script. Six different features were extracted from the Arabic character strokes. These strokes were extracted after a number of pre-processing steps, i.e., binarization, morphological dilation, diacritics removal and extraction of connected components. Identification was performed using K-Nearest Neighbours (KNN) and by using multiple distance measures within KNN such as Euclidean, Chi-squared, Manhattan, Hamming, Mahalanobis, standardized Euclidean and Minkowski. They tested their proposed system on 82 writers selected from the IFN/ENIT database and achieved a Top 1 accuracy of 90.2%.

(Ghiasi & Safabakhsh, 2013) also utilized a codebook approach for the writer identification problem. The authors realized that using connected components to identify cursive handwriting produces long and complex patters, to overcome this they proposed two novel methods of curve and line fragment extraction methods. The proposed extraction methods utilized pixel co-ordinates and angles and lengths of segments for piece wise approximation. An occurrence histogram was then utilized to generate feature vectors of each manuscript. Their proposed method was evaluated using the IAM and Firemaker datasets and achieved accuracies of 94.8% and 94.2%, respectively.

(Jain & Doermann, 2014) suggested to combine three descriptors that capture the curvature and shape of the handwritten words. They proposed to combine K-Adjacent Segments (KAS), Sped Up Robust Features (SURF) and Contour Gradient Descriptors (C). Equal weights were given to each of the descriptors and they denoted this combination as K&S&C. Fisher vectors were used for the pooling of features which were fused together using linear combination of distances. They evaluated their proposed system on the IAM, ICDAR2013 and CVL datasets and achieved an Top-1 accuracies of 94.7%, 97.4% and 99.4%, respectively.

2.1.2 Identification based on Characters/Words

Word level approaches for writer identification have also been extensively researched for many languages. Previously, these approaches would only focus on a small set of selected words for the identification and thus would only be used in a text-dependent identification mode. However, they have recently been utilized in conjunction with more advanced methods for text-independent writer identification as well.

(Zois & Anastassopoulos, 2000) focused on the single word, “characteristic” written in two languages; English and Greek. The words were first binarized and then thinned morphologically to generate horizontal projection profiles of the thinned images. Morphological operators were applied on two scales to obtain the feature vectors. Neural networks and Bayesian classifiers were used for identification purposes. Their proposed system was applied on a text dependent database of 50 writers who were asked to copy the word 'characteristic' 45 times each. An identification rate of 95% was achieved for both languages.

Similarly, (Srihari, 2003) developed a writer identification and verification system by focusing on four Latin words only; “Medical”, “Cohen”, “been” and “referred”. Gradient, concavity and structural features were extracted from these words. The classification was performed using a KNN classifier and an accuracy rate of 83% was achieved for identification while 90.9% was achieved for verification. The database used was composed of 1027 writers, who were asked to copy the above mentioned words three times each. The authors concluded that relying on full words instead of characters produce better results for identification and verification.

(Tomai *et al.*, 2004) also presented a writer identification system using a selected group of English words. 1000 writers participated in their tests, who were asked to copy the following words three times: {“To”, “Dr”, “Bob”, “Grant”, “602”, “Queensberry”, “From”, “Nov”, “300”, “Allentown”, “New”, “York”, “14707”, “Parkway”, “10”, “1999”, “Jim”, “Elder”, “829”, “Loop”, “Street”, “Apt”, “Omar”, “West”, “Virginia”}. Along with gradient, concavity and structural features, the authors also relied on the curvature shape, a word model recognizer and contour shape context as distinguishing features. KNN classifier was used achieving an accumulated performance of all words of 62% for identification and 82% for verification. The authors concluded that, longer words seem to perform better than shorter words.

Using a selected subset of words for writer identification was also applied for the Chinese script by (Zuo *et al.*, 2002). The authors used a subset of 40 Chinese words written ten times by 40 writers. Half the words were used for training while the other half was used for testing. The extracted feature vectors underwent a dimensionality reduction using Principal Component Analysis (PCA). Their proposed algorithm achieved an accuracy of 86.5% for a single word and 97.5% for a combination of ten words (sentence).

Writer identification using a selection of words in the Arabic script was performed by (Al-Ma'adeed *et al.*, 2008). Structural features such as height, length, area, distance of baseline from upper and lower limits of words were used along with features such as edge directions at multiple angles and moment invariants. They tested their proposed system on a database of 100 writers who wrote twelve common Arabic words and four common Arabic phrases, copied twenty times. 75% of the database was used for training while the remaining 25% was used for testing. Only the Top 10 results were reported where an accuracy of 90% was achieved using a KNN classifier.

(Fiel & Sablatnig, 2012) proposed a writer identification and retrieval system based on the codebook approach. Rather than generating the codebook using graphemes or textural based identifiers, the authors proposed to extract the Scale Invariant Feature Transform (SIFT) features from all writers after extracting words as connected components from the handwritten text. These features were clustered and a codebook was generated. By using SIFT, the authors avoided the binarization step hence eliminating the associated problems of poor binarization of faded text or low contrast documents which can lead to a loss of important identifying features. Their proposed system achieved an accuracy of 91% using 650 writers of the IAM database. Later (Fiel & Sablatnig, 2013) used local SIFT approach for identification where SIFT features were used to create a visual vocabulary by a clustering process using Gaussian Mixture Models (GMM). This enabled the authors to calculate the Fisher vector for each image. Classification was performed using the least distance rule. Their proposed system was applied on the CVL and ICDAR2011 databases in which they showed top 1 results of 97.8% and 91.3% respectively.

The bag of features approach was utilized by (Hu *et al.*, 2014) for writer identification in Chinese. The authors realized that codebook generation with the help of graphemes was not ideal for a language with such complicated structures. They instead proposed to utilize SIFT descriptors in place of graphemes for training the codebook. Two recent coding methods: Locally Constrained Linear Coding (LLC) and Improved Fisher Kernels (IFK), were used for encoding each SIFT descriptor. They evaluated their proposed method on the CASIA offline dataset (240 writers) and achieved Top 1 accuracies of 96.25% and 95.42% using the IFK and LLC methods, respectively.

(Jain & Doermann, 2013) proposed a method that replicates the approach taken by forensic experts for writer identification. The authors proposed a novel contour gradient feature that captures the shape and curvature of the characters and paired it with character segmentation. These two features were clustered to achieve a pseudo-alphabet of each writing style. A distance measure between two pseudo-alphabets was used for classification. They evaluated their proposed method on the IAM (301 writers) and Arabic-MADCAT (316 writers) datasets to achieved Top 1 accuracies of 96.5% and 87.5%, respectively.

Wu et al. (Wu *et al.*, 2014) proposed to perform writer identification using two features; SIFT descriptors (SD) and their corresponding scales and orientations (SO). These two features, extracted from the segmented words of the writers, were used in different ways during the identification process. The extracted SDs from the writers were clustered in order to generate a training codebook. This was followed by an enrolment stage where the SDs of the training images were converted, with the help of the SD codebook, into an SD Signature

(SDS). At the enrolment stage a scale and orientation histogram (SOH) was also generated from the extracted SOs. Identification was performed by extracting and comparing SDS and SOH of the query images with the enrolled ones. Thus resulting in two matching distances, which were fused to get the final identification result. Their presented system achieved a Top 1 accuracy of 98.50% on the IAM dataset (650 writers), 92.40% on the Firemaker dataset and 95.20% on the cropped version of the ICDAR2011 dataset.

In the same year, (Tang *et al.*, 2014) proposed a modification of two structural features: The SD was modified to include orientation information and called modified SIFT descriptor (MSD) and a Triangular descriptor (TD) that was a relationship between three selected fixed points. A codebook was generated using the extracted MSDs and the bag of words approach was utilized to generate MSD Histograms (MSDH). Similarly, a TD Histogram was generated by tracking contour points of the handwritten words. Classification was performed by considering the distance between the MSDHs and the TDHs. Their proposed system was evaluated on the IAM and ICDAR2013 datasets and achieved an accuracy of 97.1% (657 writers) and 95.2% (250 writers), respectively.

2.1.3 Identification based on Lines

From the perspective of human thinking, line level identification seems more effective than character or grapheme counterparts. Using lines, one apparently extracts more information to work with, which should improve the identification. This section focuses on works where line level identification has been explored.

(Marti *et al.*, 2001) proposed a writer identification system by extracting twelve local features from text lines. These twelve local features were derived from the width, the slant and the zones of the handwritten text. They applied their proposed system on the IAM database using only twenty writers. Two different classifiers were used: KNN and Artificial Neural Networks (ANN). An accuracy of 87.8% and 90.7% was achieved using KNN and ANN, respectively.

(Hertel & Bunke, 2003) also proposed a line level writer identification system where features were extracted from individual text lines. The features were based on the blobs inside the ink loops, the distances between the upper and lower contours and the distance between the connected components. Fractal features were also utilized which observed the growth in area (pixels) of handwritten text after applying a thinning operation. The KNN classifier was used for identification purposes. They applied their proposed system on 50 writers from the IAM database and achieved an accuracy of 90.7% for a single line and 99.6% for a combination of lines.

Line based writer identification for Persian script was carried out by (Rafiee & Motavalli, 2007). Eight features, which were based on the width and height of the words within each text line, were extracted. They applied their proposed system on a database of twenty writers, where each writer was asked to copy a line of text twice followed by three to five arbitrary lines of text. An accuracy of 86.5% was achieved using Neural Networks (NN).

(Kırlı & Gülmezoğlu, 2012) presented an offline text independent writer identification system by extracting global and local features from segmented lines of handwritten text. For local feature extraction, each text line was divided into three writing zones and a dynamic window was used to extract six local features. Along with these, global features such as slant, thickness, width, surface area, density and size were also extracted. KNN, GMM and Normal Density Discriminant Function (NDDF) Bayes classifiers were used to test individual feature performance along with the performance of combination of different features. They applied their proposed system on the IAM dataset using 93 writers and a performance of 98.76% was reported. They also concluded that for their feature set the NDDF Bayes classifier outperformed KNN and GMM.

2.1.4 Identification based on Paragraphs/Full Text

Along with character level and text line level writer identification, researchers have also explored extracting features from paragraphs or full page text for writer identification. Such methods consider the handwriting as a texture and thus identification as a textural problem.

(Schlapbach & Bunke, 2004) used an HMM (Hidden Markov Model) to identify the unknown images. For each writer, the authors built a single HMM recognizer using the features extracted from a shifting pixel-wide window, the sliding window which extracts 9 features in total with three global and six local. The global features included the number of black pixels in the window, the second order moment and the centre of gravity. While the local features extracted were the positions of the top most and lowest pixel, the fraction of black pixels between these two limits and the number of black to white transitions. Using this 9-dimensional feature vector the corresponding HMM is trained for every writer and the authors were able to achieve a 96.5% accuracy using 100 writers from the IAM database. The identification was achieved using a log-likelihood score to rank the writers. The same authors proposed a further improvement to their previous work in (Schlapbach & Bunke, 2006) where they replaced the HMM with a GMM. At the time GMM was used mainly in the speech recognition community but by applying the same concept to writer identification, the authors were able to achieve an improved performance. When compared against their previous result of 96.5%, an identification rate of 98.4% was achieved using 100 writers from the IAM database. Furthermore, GMM was conceptually simpler and faster to train than the HMM models. A drawback of both these systems was that they were highly dependent on perfect line segmentation to achieve the desired results as poor segmentation would greatly affect the performance of the system.

(Shahabi & Rahmati, 2006) proposed a text independent writer identification system using a full A4 page. Each page was divided into four sections, three sections were used for training while the last section was used for testing. The documents were pre-processed through a binarization process, normalization of space between words and smoothing of the text. This was followed by feature extraction using multi-channel Gabor filters. Euclidean distance and chi-square distance were used for classification. Their system was tested on a text independent database of 25 writers and the system achieved a Top 1 accuracy of 88% and a Top 3 accuracy of 92%. The same authors further proposed a text dependent writer identification and verification system in (Nejad & Rahmati, 2007). They used a text dependent database containing 40 writers to test their system. They performed identification at word level as well as paragraph level. For word level identification they achieved an accuracy of 45% whereas for full text level identification they achieved an accuracy of 82.5%. Helli and Maghoddam (Helli & Moghaddam, 2008) also utilized Gabor filters for the purpose of feature extraction and extended the concept by using a Longest Common Subsequence (LCS) based classifier for the purpose of identification. They applied their system on the PD100 database using 100 writers and achieved an accuracy of 95%. The same authors extended their previous work by using a Feature Relation Graph (FRG) to represent the Gabor features (Helli & Moghaddam, 2010). The FRG was constructed for each writer using a fuzzy method. Identification was performed using a graph similarity approach which achieved an identification accuracy of 100% when tested on the PD100 database using only 80 writers.

(Al-Dmour & Abu Zitar, 2007) proposed a page level writer identification system by selecting an ideal subset of features from multi-channel Gabor filters, spectral statistical measures and grey level co-occurrence matrices. They used a database of 20 writers, where each writer was asked to copy an A4 document of text twice. One document per writer was used for training while the other was used for testing. The authors used a number of classifiers to test their system, including Linear Combined Distance (LDC), KNN, SVM and Weighted Euclidean Distance (WED). Identification accuracy of 90%, 47%, 57% and 69% were achieved for LDC, WED, KNN and SVM respectively.

Siddiqi and Vincent (Siddiqi & Vincent, 2008) proposed a system which utilized both the global and local features of handwriting. They proposed a two-step approach: (i) first global features were extracted by localizing the oriented segments using Gabor filters and by using a direction criteria the handwriting class was determined, (ii) the query document was identified using local features by searching only in the specific class that was determined before. This was made possible by clustering the frequent patterns of handwriting from every writer. Classification was made possible by comparing the patterns using a Bayesian classifier. They tested their proposed system on the IAM database using 100 writers and reported an accuracy of 92%. The same authors later proposed extracting identifying information from the contours of handwriting at local and global levels (Siddiqi & Vincent, 2009). Their proposed system reported an accuracy of 86% when applied on the full IAM database composed of 650 writers. The same authors further improved their previous work in (Siddiqi & Vincent, 2009) by extracting information from the orientation and curvature of the chain code sequence of handwriting. This system improved their previous result when applied on the full IAM database of 650 writers by producing an accuracy of 89%.

Brink et al. (Brink *et al.*, 2012) demonstrated the importance of width patterns in writer identification by proposing the QuillHinge feature to extract identifying characteristics from the direction of ink traces as well as the width of ink strokes. Their proposed system showed an accuracy of 86% when applied on the Firemaker database (lower case) and 97% when applied on the full IAM database of 650 writers.

(Bertolini *et al.*, 2013) considered the handwriting text as a texture and used Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) to extract textural features for writer verification and identification. They built upon a previously reported work using the dissimilarity framework approach by (Hanusiak *et al.*, 2012) and extended the idea to writer identification. The concept underlying the dissimilarity framework is based on the mapping of texture vectors into dissimilarity vectors where two classes only can be constructed: positive population and negative population. The samples from both classes are then used to train a binary SVM classifier. Given a reference text image and a query text image, the system calculates the corresponding dissimilarity vector and uses the trained SVM to classify it (this is to verify whether the query image is from the same class as the reference image). In writer identification, the query image is compared with all images in the database to extract the corresponding dissimilarity vectors. Each dissimilarity vector is then classified with SVM and the hits found for each writer according to SVM are combined via a fusion function (sum, max, median, product, etc.) to determine the closest writer. The system was tested using two databases: the Brazilian Forensic Letter (BFL) and the IAM database (650 writers). An accuracy of 99.2% and 96.7% was achieved on the BFL and IAM database, respectively.

(Djeddi *et al.*, 2013) proposed a writer identification system that would support multiple scripts since the authors argued that the writing style of an individual remains constant irrespective of the script used. Furthermore, they intended to demonstrate that their system would train even on a very limited amount of text and thus be more effective in real world scenarios. Gray Level Run Length (GLRL) matrices were used for the purpose of feature extraction and KNN and SVM were used for classification purposes. The ICFHR2012 dataset (English and Greek from 126 writers) was used to test their proposed system. Using the English text accuracies of 87.30% and 82.54% were achieved using the SVM and KNN classifiers, respectively whereas using the Greek text, accuracies of 88.09% and 91.27% were achieved using the SVM and KNN classifiers, respectively.

(Newell & Griffin, 2014) proposed to use two texture based descriptors for the purpose of writer identification. The first of these methods is the oriented Basic Image Feature (oBIF) which encodes six different Difference of Gaussian (DoG) responses as histograms. The authors further improved this by proposing a delta encoding scheme which takes into account the lexical content of text. The delta encoding scheme determines the mean oBIF histogram from all the writers of the training set and then subtracts that from the oBIF histogram of the query document. The delta coding scheme was shown to be the best performing system in the ICFHR2012 competition. The authors also evaluated their system on the IAM dataset (301 writers) and achieved an accuracy of 99%.

By observing the effectiveness of texture descriptors in classification and the discriminatory strength of fragments in writer identification problems, (Hannad *et al.*, 2016) proposed a writer identification system by extracting textural information from small fragments of text. Handwritten documents were first segmented into small fragments using a window size of 100 x 100 and then fed to three feature extractors: LBP, LTP and LPQ. The authors demonstrated that out of the three descriptors, LPQ gave the best identification results and maintained its accuracy as the number of writers was increased. The handwritten documents were compared and classified by using a simple dissimilarity measure. Their proposed system was evaluated using the IAM (657 writers) and IFN/ENIT Arabic (411 writers) datasets which produced Top 1 accuracies of 89.54% and 94.89%, respectively.

2.2 Datasets Used

Like other scientific domains, the hand writer identification community has put together a large number of datasets in various languages, that can be used to evaluate and compare different systems developed for writer identification. Table 2.1 gives an overview of the various datasets available in different languages.

The proposed systems presented in this thesis were evaluated using six publicly available

Dataset	Year	Language	Content	Writers
NIST (Wilkinson *et al.*, 1992)	1995	English	Isolated digits	3600
Al-Isra (Kharma *et al.*, 1999)	1999	Arabic	Sentences	500
Firemaker (Schomaker & Vuurpijl, 2000)	2000	English	Paragraphs	250
GRUHD (Kavallieratou *et al.*, 2001)	2001	Greek	Text, symbols	1000
IAM (Marti & Bunke, 2002)	2002	English	Sentences	657
IFN/ENIT (Pechwitz *et al.*, 2002)	2002	Arabic	Words	411
AHDB (Al-Ma'adeed *et al.*, 2002)	2002	Arabic	Sentences, checks	100
ARABASE (Amara *et al.*, 2005)	2005	Arabic	Sentences, words	400
CENPARMI-A (Alamri *et al.*, 2008)	2008	Arabic	Words, characters	328
RIMES (Grosicki *et al.*, 2008)	2008	French	Sentences	1300
HCL2000(Zhang *et al.*, 2009)	2009	Chinese	Characters	1000
ADAB (Abed *et al.*, 2011)	2011	Arabic	Words	170
SCUT-COUCH (Jin *et al.*, 2011)	2011	Chinese	Characters	190
CASIA (Liu *et al.*, 2011)	2011	Chinese	Text, characters	1020
ICDAR2011(Louloudis *et al.*, 2011)	2011	Englis, French, German, Greek	Sentences	8+26
AHTID/MW (Mezghani *et al.*, 2012)	2012	Arabic	Text lines	53
KHATT (Mahmoud *et al.*, 2012)	2012	Arabic	Sentences	1000
QUWI (Al Maadeed *et al.*, 2012)	2012	English / Arabic	Sentences	1017
ICFHR2012 (Louloudis *et al.*, 2012)	2012	English, Greek	Sentences	26+100
CVL (Kleber *et al.*, 2013)	2013	English / German	Sentences	311
HaFT (Safabaksh *et al.*, 2013)	2013	Farsi	Sentences	600

Table 2.1: An overview of the different datasets available for hand writer recognition.

datasets to demonstrate the performances. Of the six datasets, three are English: IAM[1] (Marti & Bunke, 2002), Firemaker [2] (Schomaker & Vuurpijl, 2000) and CVL[3] (Kleber *et al.*, 2013), two are Arabic: AHTID/MW[4] (Mezghani *et al.*, 2012) and IFN/ENIT[5] (Pechwitz *et al.*, 2002) and one is of hybrid-language i.e. ICDAR2011[6] dataset (Louloudis *et al.*, 2011). The details of the datasets used are mentioned below.

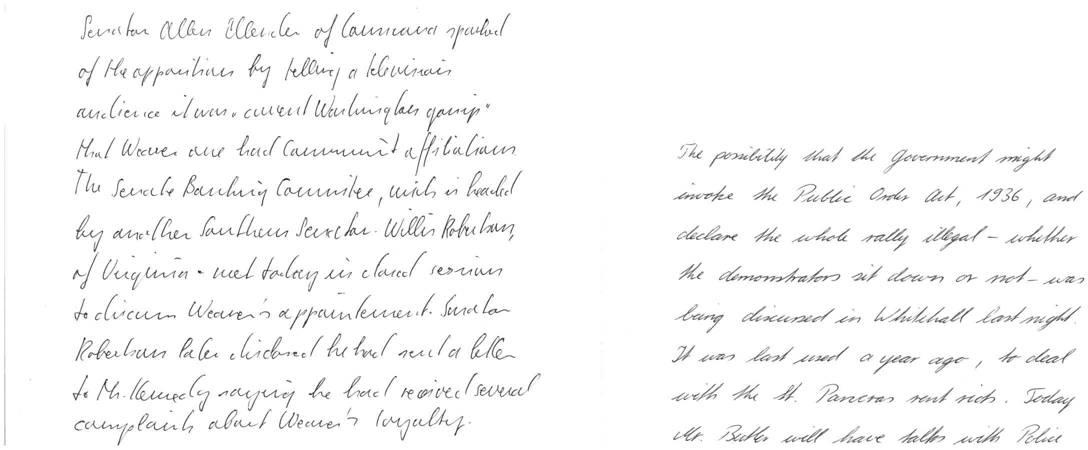
2.2.1 IAM Dataset

The IAM dataset is the most widely used English database in the field of writer identification. Handwritten samples from 657 writers have been collected in this database, all of which are scanned at 300 dots per inch (DPI) and saved in greyscale format. Out of the 657 writers 301 have contributed two or more than two handwritten documents while the rest of the 356 writers have contributed only a single handwritten document. For the evaluation performed in Chapter 4, this dataset was arranged according to the 100 writer identification task, whereas for the remaining Chapters, this dataset was arranged as proposed in (Bulacu & Schomaker, 2007) where every writer is limited to a maximum of two sample; one for training and the other for testing. For writers that have contributed multiple documents, only two are retained whereas the writers that contributed only a single document, have that document split roughly in half. One half is used for training while the other half is used for testing. Using this arrangement, 650 writers with usable data are left. Figure 2.3 shows a sample of handwritten text from the IAM dataset.

2.2.2 IFN/ENIT Dataset

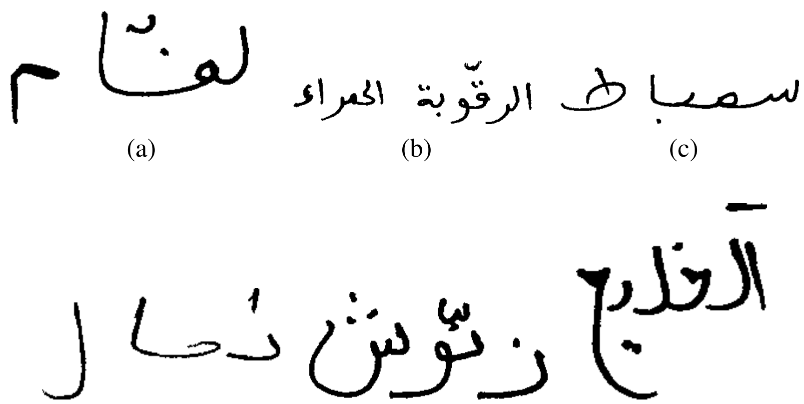
The IFN/ENIT can be considered to be the most widely used Arabic database for the purpose of hand writer identification. It consists of 411 writers who have contributed a total of 26,000 handwritten samples of different Tunisian village names. All samples are scanned at 300 DPI and are saved in binary format. For our investigation, this dataset has been arranged as explained in (Hannad *et al.*, 2016), where a significant smaller number of samples per writer are used in order to simulate conditions of the real-world scenarios. Under this setting, for every writer, 30 and 20 randomly selected words are used for training and testing purposes,

[1] http://www.iam.unibe.ch/fki/databases/iam-handwriting-database
[2] http://www.ai.rug.nl/~lambert/overslag/67bebf61751444b8630f3f/firemaker.tgz
[3] http://www.caa.tuwien.ac.at/cvl/category/research/cvl-databases/
[4] http://ieeexplore.ieee.org/document/6424426/
[5] http://www.ifnenit.com/
[6] http://ieeexplore.ieee.org/abstract/document/6065553/



(a) (b)

Figure 2.3: Samples of handwritten text from IAM dataset written by different writers.



(a) (b) (c)

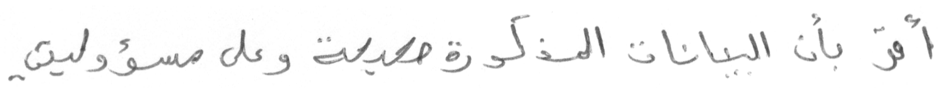
(d) (e) (f)

Figure 2.4: Samples of handwritten text from IFN/ENIT dataset written by different writers.

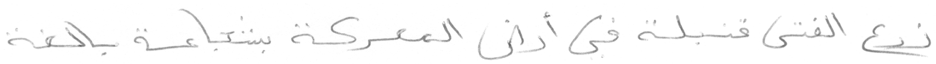
respectively. Figure 2.4 shows samples of handwritten text from the IFN/ENIT dataset.

2.2.3 AHTID/MW Dataset

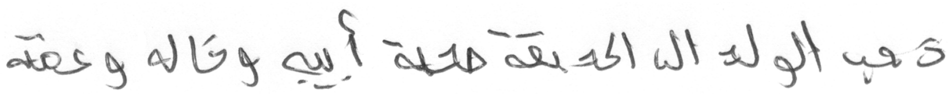
AHTID/MW dataset consists of handwritten samples from 53 native Arabic writers, who are of varying ages and varying educational backgrounds. These writers have contributed a total of 3,710 text lines without any restriction on the type of pen being used. All the writers have contributed a total of 126,511 characters, which seem to cover all forms of the Arabic characters, i.e. the isolated, beginning, middle and end form of writing an individual Arabic character. All samples are scanned at 300 DPI and saved in grayscale format. All of the text samples have been divided into 4 sets out of which 3 sets have been used for training and the last set has been used for testing. Identification results are obtained after four-fold cross



(a)



(b)



(c)

Figure 2.5: Samples of handwritten text from AHTID/MW dataset written by different writers.

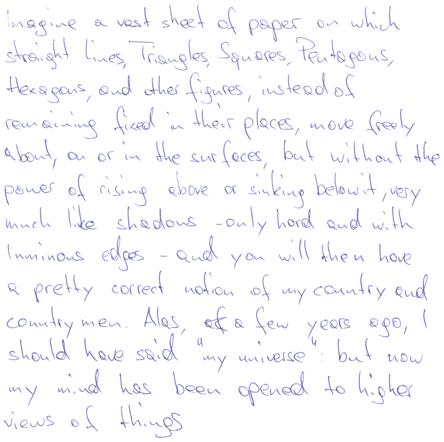
validation. Figure 2.5 shows samples of handwritten text from the AHTID/MW dataset.

2.2.4 CVL Dataset characters

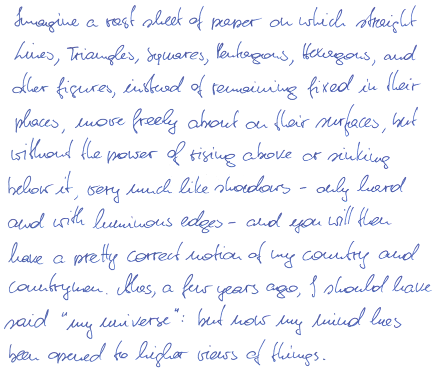
The CVL dataset contains handwritten documents from 311 writers. Of these 311 writers, 27 writers produced 7 documents of text while the remaining 284 writers produced 5 documents of text. Each writer produced one sample in cursive German and the rest in cursive English. All documents are scanned at 300 DPI and saved in a colour depth of 24 bit format. In our experiments, we have only utilized the English documents thus leaving us with only four documents per writer. Out of these four, three documents per writer are used for training, while the fourth one is used for testing. Figure 2.6 shows samples of handwritten text from the CVL dataset.

2.2.5 Firemaker Dataset

The Firemaker dataset contains handwritten samples taken from 250 writers where each writer contributed 4 pages of handwritten text. In page 1 the subjects were asked to copy five short paragraphs using their own normal handwriting. In page 2 the subjects were asked to do the same for two paragraphs but only using uppercase handwriting. In page 3 the subjects were encouraged to write in a "forged" text; to write in a style that is not their own. On page 4 the subjects were asked to explain a given cartoon in their own words. As was done in (Bulacu & Schomaker, 2007) only page 1 and page 4 were used in our experiments. Figure 2.7 shows samples of handwritten text from the Firemaker dataset.

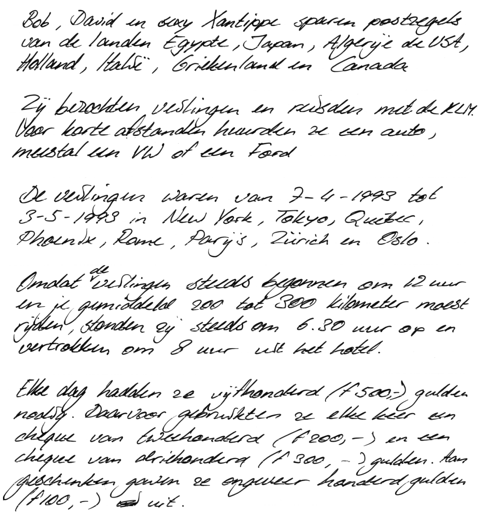


(a)



(b)

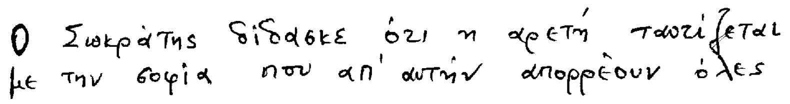
Figure 2.6: Samples of handwritten text from the CVL dataset written by different writers.



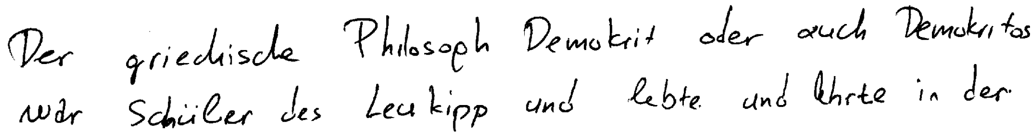
(a)

(b)

Figure 2.7: Samples of handwritten text from the Firemaker dataset written by different writers.



(a)



(b)

Figure 2.8: Samples of handwritten text from the ICDAR2011 dataset written by different writers.

2.2.6 ICDAR2011 Dataset

The ICDAR2011 dataset contains pages of handwritten text written in four different languages: English, French, Greek and German. 26 writers have contributed to this dataset by writing 2 full pages of text for each language, thereby giving 8 pages of text per writer. A variation of the ICDAR2011 dataset is known as the ICDAR2011 cropped dataset which is made by cropping only the first two lines from every image. This significantly reduces the available text for each writer. The cropped variation of the dataset was used in our experiments. For every writer 5 images are used for training while 3 images are used for testing. Figure 2.8 shows samples of handwritten text from the ICDAR2011 dataset.

2.3 Conclusion

This chapter has reviewed previous state of the art systems of writer identification. The works carried out were classified into either grapheme level, word level, line level or textural level approach. From the literature review, it can be identified that the grapheme and codebook approaches have been widely researched. A universal grapheme codebook, which helps in generating occurrence histogram descriptors provide more than acceptable results for various scripts on which it has been applied. Although universal codebooks reduce computational costs, they have the problem of needing to be re-generated if the script changes. Word level approach has only recently been used effectively for text-independent writer identification and has been shown to be very effective when used in conjunction with another feature extractor such as SIFT. Very limited research went into identification of writers based on full lines of text. Although for us humans it does make more sense to identify based on lines of text, it appears that for computers the more effective way of identification is on a grapheme or textural level. Considering handwriting as a texture and identification as a textural problem provides the most promising results for larger datasets of writers. Producing high accuracy identification when a large number of writers are involved is what is being researched nowadays, as in a real world setting a system is expected to perform well, irrespective of the number of writers. Texture level identification has shown to be robust to the number of writers and is the level at which writer identification is being researched at the most. Based on these findings, in this thesis the writer identification problem was explored using only the most promising approaches and was tackled at a textural level, multi-scale textural level and at word level combined with additional feature extractors. The results shown in Chapter 4, 5 and 6 demonstrate that the more recent approach of pairing word level segmentation with a strong feature extractor provides the best accuracy for large datasets.

The next chapter presents a novel method for efficiently segmenting overlapping subwords within hand written Arabic words. Identification of writers in Arabic text is considerably more challenging than that of those in other scripts. This is largely due to the style in which Arabic script is written, where an overlapping approach is used to make maximum use of the writing space available. Segmentation of these Arabic overlapping characters proves to be a considerable challenge and is the contributing factor in producing poor identification results.

Chapter 3

Efficient Segmentation of sub-words within Handwritten Arabic Words

3.1 Introduction

The Arabic language is used by more than 300 million people in over 20 countries (Mezghani *et al.*, 2012). The Arabic writing style is also used by many other languages such as Persian, Urdu, Pushto and other regional languages of Pakistan, Afghanistan and Iran. Following Latin script, it is the second most widely used script in the world. The Arabic writing style is cursive both in printed as well as hand written text and is written from right to left. It consists of 28 letters but these letters change their shape based on their location within the word as shown in Figure 3.1. Most of the letters will be written differently depending on whether they are located at the beginning, middle, end or alone in the main word. Thus, an Arabic word is composed of many sub-words known as Part of Arabic Word (PAW).

Segmentation can be defined as the process of dividing or separating an image into smaller segments or into useful regions based on some conditions. Segmentation is considered as a core step for any recognition or classification method. For the text within a document to be effectively recognized it must be segmented accurately. Segmentation of handwritten text

غ	غـ	ـغـ	ـغ
Isolated form	Beginning form	Middle form	Ending form

Figure 3.1: An Arabic letter written differently based on its position within the word (Mezghani *et al.*, 2012).

unlike printed text is considered to be a very challenging and difficult problem. A system that showed an accuracy of 99.55% on the ICDAR 2009 printed Arabic text dataset using a ratio metric of detection and recognition, achieved merely 56.1% accuracy when it was applied on a dataset of Arabic handwritten text (Manohar *et al.*, 2011). The reason for this sharp drop in accuracy is due to the challenges presented by hand written text such as touching and over lapping components, irregularities in the skew of different lines etc.

In handwritten documents the property of a Connected Component (CC) such as area, height, width etc. can be a very reliable distinguishing factor. A handwritten note written by the same writer very rarely has words of varying dimensions. The writer tries his or her best to maintain the general dimensions of the text. At most, if anything does change it is the skew of text lines but not the word's width, height or area. And fortunately, the proposed method is not affected by skew of the text lines.

The Arabic writing style is very different from other languages. For example English and many other similar languages have a "one-letter-after-another" guideline that is strictly followed in printed as well as handwritten documents but Arabic words follow a mixed and somewhat jumbled writing style where many letters are written before the previous one is finished. Basically, Arabic handwriting is very compact and any free space is utilized. This is the reason for the jumbled letters written on top of one another. For successful recognition and classification of the Arabic words all the letters of every word must be segmented properly.

Quite a few detection approaches have been proposed in literature for Arab writings, and among them the projection profile analysis is the more commonly used algorithm (Margner & Abed, 2012). However, the problem with the projection profile method is that it does not work properly with Arabic language and the results further deteriorate in case of hand written Arabic text. When the projection profile method is used to separate the individual connected components or part of Arabic words a lot of the over lapping letters are left out and do not appear in the segmented results. Figure 3.2 shows the overlapping/stacking problem of the Arabic language.

The letters within the red markers are left out during the projection profile method because these letters are hidden by the larger connected components. This is valuable information that is lost and can be a deciding factor in the classification of the writer.

In this chapter, a new sub-word segmentation method is proposed (Khan *et al.*, 2014). The novelties of the presented system are that (i) it does not depend on the skew of the document or the skew of the individual lines, (ii) it is very robust in a way that the proposed method is not affected by distortion and (iii) it allows for the individual segmentation of various 'groups' within the documents e.g. characters can be segmented with or without their respective dia-

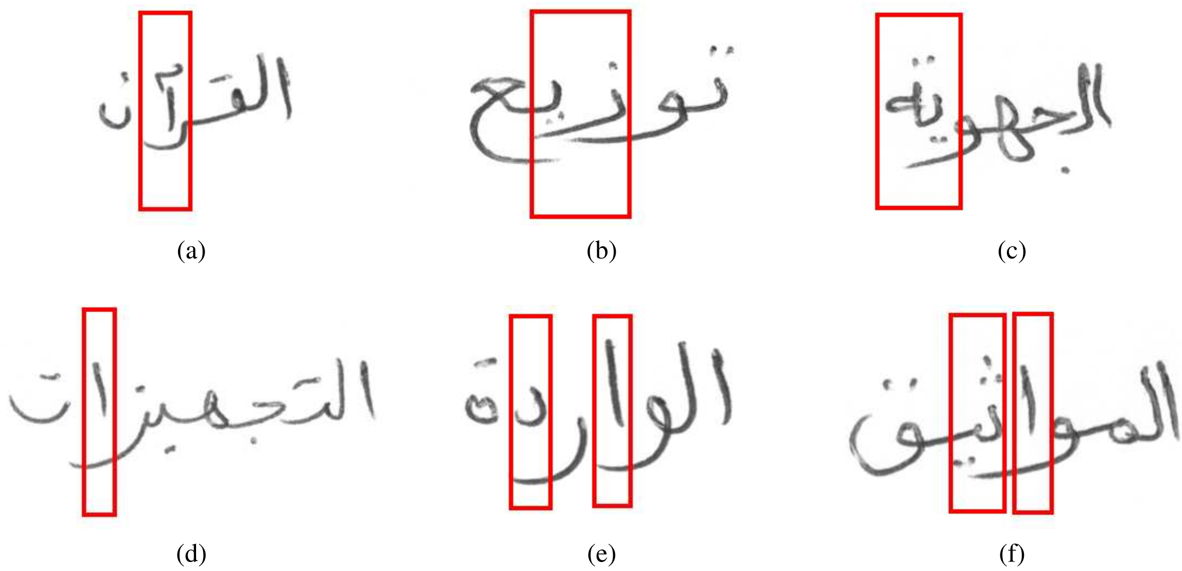


Figure 3.2: Overlapping of characters in Arabic words

critics.

This chapter is organized as follows: Section 3.2 details the process of the presented system. Section 3.3 shows the experimental results of this system. And finally Section 3.4 outlines the conclusions.

3.2 Proposed System

Pre-processing just may be the first step for almost every segmentation method and this is no exception for our proposed system. In this system, segmentation is achieved directly as a result of pre-processing i.e. binarization. Binarization is a very popular and basic pre-processing step where a colour or greyscale image is converted to logical format. Otsu's method (Otsu, 1979)can be used to binarize an image as it accomplishes the task with minimal loss of information and by producing minimum noise.

When binarizing an image, using too low a threshold will split the large connected components and remove small ones whereas using a very high threshold will cause a thickening of the connected components (Dinges *et al.*, 2013). Based on this principle it was determined that if binarization was applied to text at minutely changing thresholds, small components will thicken and become more prominent at high thresholds while disappear or become negligible at low thresholds.

Binarization produces a lot of noise especially at low thresholds. This noise is produced

(a) (b)

Figure 3.3: (a) Before dilation (b) After dilation

even within the regions of interest and greatly distorts and affects the collective graph of all the threshold levels. Furthermore, in handwritten Arabic text a lot of sub-words are not connected properly and have very small gaps between them, thus unintentionally converting one sub-word into two. To overcome these problems morphological dilation (equation 3.1) is performed using a very small structuring element at low thresholds.

$$A \oplus B = \left\{ z | \left(\hat{B}\right)_z \cap A \neq \emptyset \right\} \tag{3.1}$$

Where A represents the word or image and B represents the structuring element. Figure 3.3 shows the before and after result of dilation

Binarization of an image can be represented mathematically by equation 3.2

$$g(x,y) = \begin{cases} 1 & if\, f(x,y) > T \\ 0 & otherwise \end{cases} \tag{3.2}$$

Where T is the threshold at which the image is binarized and *g(x,y)* is the binarized image obtained. This threshold, T is incremented or decremented in small steps to produce a new output, *g(x,y)* for every increment i.e. $T = \{0, 0.001, 0.002\ldots, 1\}$.

The resulting image *g(x,y)* for every threshold is processed and its connected components (CC) found and labelled. Properties such as height, width, area etc. are calculated for these connected components. Here, we chose to use areas of the connected components but any property can be used. These detected properties are plotted onto a graph where the y-axis represents that selected property which in this case is area of the CC and the x-axis represents the number of sub-words or connected components in the word.

For example, the same word shown in Figure 3.3 would be used. Now if binarization is applied and the connected components or sub-words within it are measured and plotted then the graph shown in Figure 3.4 would be the area representation of the sub-words at a single threshold. The points higher up on the y-axis represent sub-words whereas the points lower down (usually found to be below 200) on the y-axis represent the diacritics. Now if the process of binarization is repeated roughly a thousand times with minutely changing thresholds and

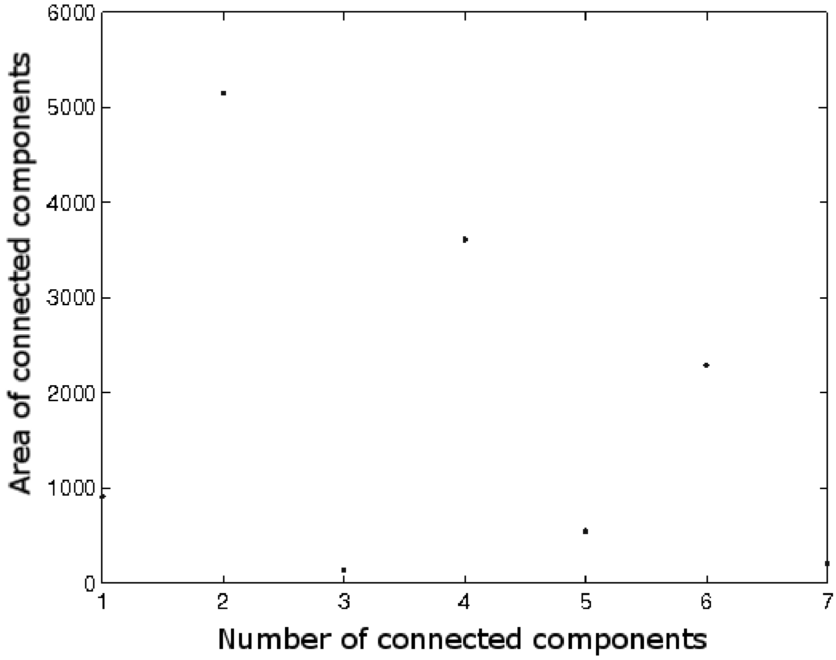


Figure 3.4: Connected components at a single threshold

the results are plotted over the same graph (equation 3.3), the connected components that were represented by dots before become distinguishable clusters. These clusters are shown in Figure 3.5.

$$\sum g(x,y) \qquad \forall \quad k \tag{3.3}$$

In equation 3.3, $g(x,y)$ represents a single binarized image and k represents the threshold level.

In this cluster form, it becomes much easier to distinguish between each connected component. The clusters grow based on the size of the sub-word. The bigger the sub-word the bigger its cluster and vice versa. The clusters representing diacritics do not show much growth. This also helps in safely removing the diacritics from the segmented results.

At times, there are words that have letters, having the same dimensions, represented more than once. This causes the repetition of clusters on the graph. For example, for a word having the letter 'alif' written with the same dimensions, but written twice will produce two different clusters in the same are region as shown in Figure 3.6.

To avoid this problem and to help with the process of automatic segmentation, the clusters are forced onto a single point on the x- axis which causes the repeated clusters to be merged

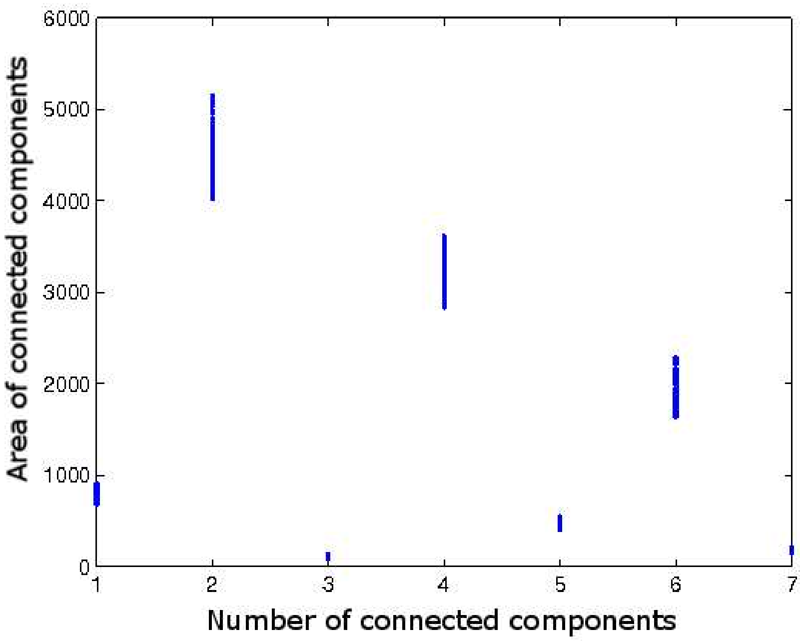


Figure 3.5: Clusters representing sub-words

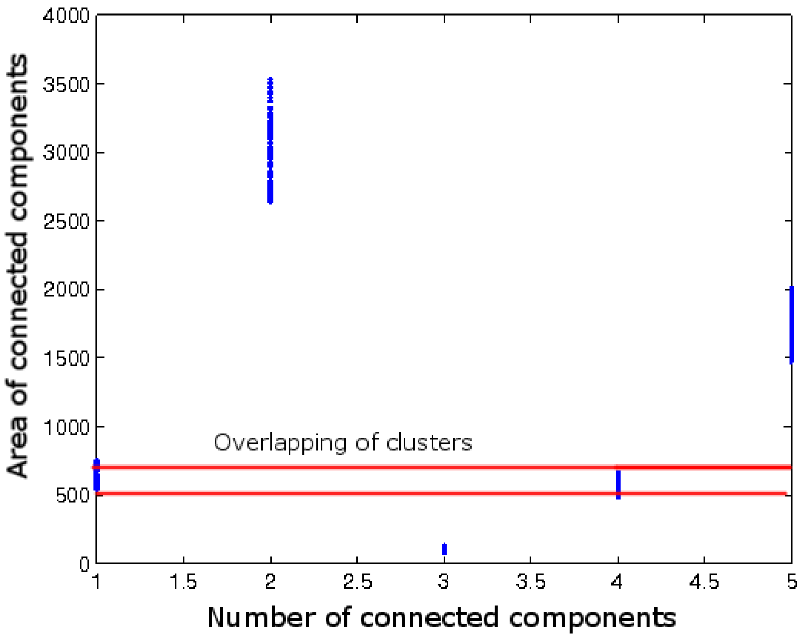


Figure 3.6: Overlapping of clusters in the same area region

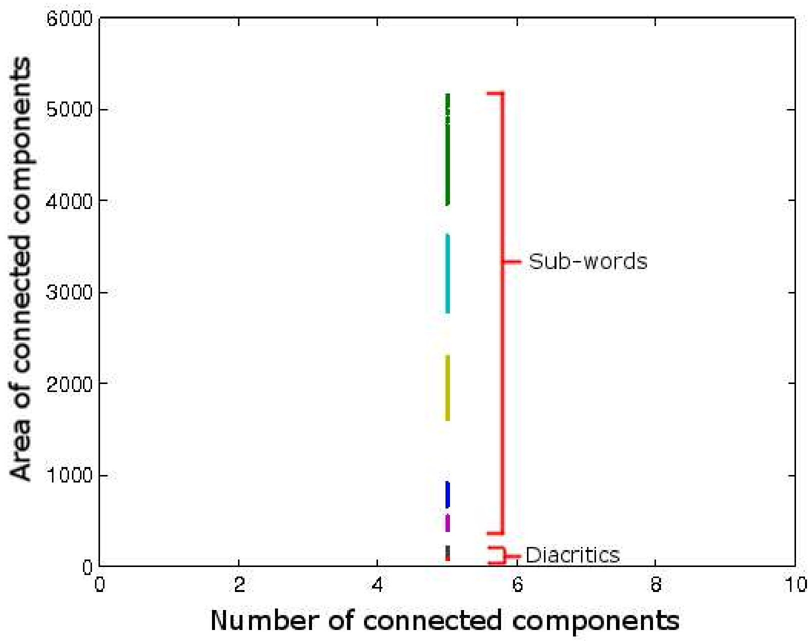


Figure 3.7: Clusters merged together. Lowest clusters represent diacritics, higher clusters represent sub-words

into each other as shown in Figure 3.7.

By looking at the clusters and the data that generated these clusters, one can state that as per the central limit theorem the data in the above plot is normally distributed. The central limit theorem states that if the data is made up of a large number of independent random variables that have a well-defined variance, then this data will follow a normal distribution (Johnson, 2004). Suppose that $\{X_1, X_2, X_3 \dots X_n\}$ represents the data for a single cluster in the above plot, which are independent and identically distributed random values. Now for a large enough n the distribution will converge to a normal distribution, mathematically given by equation 3.4.

$$\sqrt{n}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - \mu\right) \xrightarrow{d} N\left(0, \sigma^2\right) \tag{3.4}$$

Where σ^2 is the variance and μ is the mean.

Now that the normal distribution has been established, character segmentation can be performed automatically from the above plot by letting a decision based algorithm choose, identify and label the clusters of interest while ignoring the noise and unimportant points/clusters. This automatic recognition is accomplished using a 95% confidence interval (Mendenhall

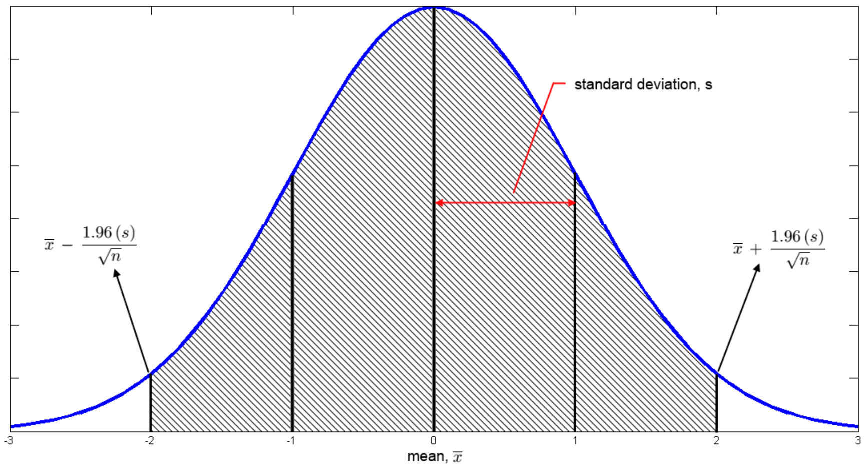


Figure 3.8: Normal distribution curve representing a cluster. The filled area represents the 95% confidence interval, whose limits are calculated using equation 3.5

et al., 2012). A 95% confidence interval gives the probability that the true mean of the distribution will lie between two set values, or between the calculated limits. These limits help to automatically select the clusters of interest.

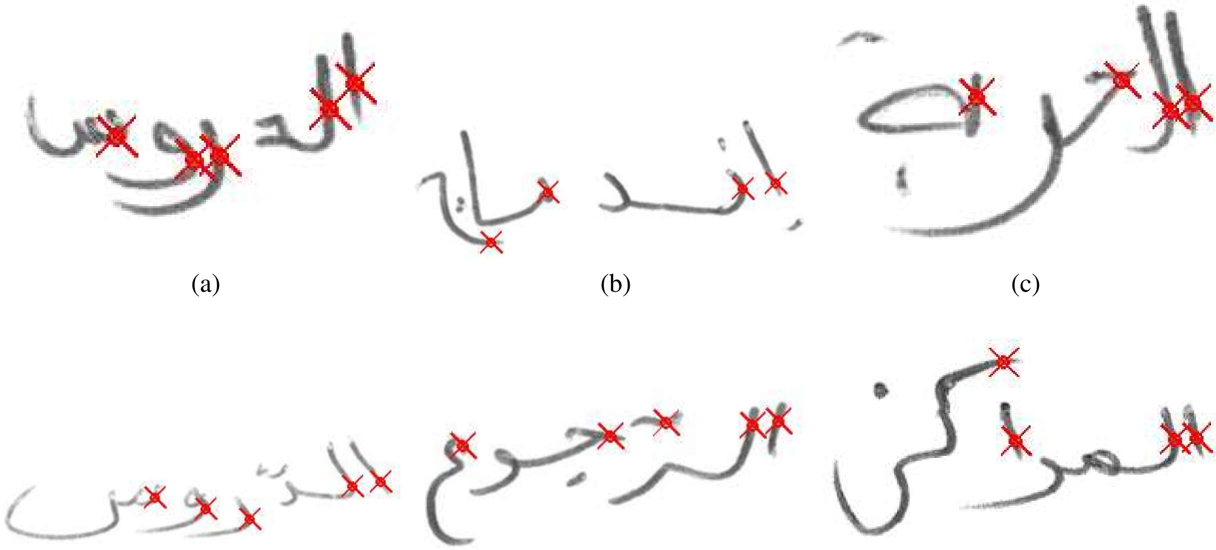
Being normally distributed each cluster can be represented by a normal distribution curve as shown in Figure 3.8. The 95% confidence interval corresponds to covering 95% of the area of the distribution curve, thus making the probability of any value outside of this area to be less than 5%. Calculating the limits of this area under the distribution curve will correspond to the start and end points of the clusters. In this way, automatic segmentation of each cluster of interest is achieved by calculating the 95% confidence interval limits. These limits are calculated using equation 3.5.

$$\bar{x} \pm \frac{1.96\,(s)}{\sqrt{n}} \tag{3.5}$$

Where $\bar{x}$ is the mean, s is the standard deviation and n is the sample size.

3.3 Experimental Results

To test the effectiveness of the proposed system, 537 randomly selected words from the AHTID/MW dataset were used. The random selection was done using a random number generator which would select the writer first and then randomly select the word written by that writer. Each word tested was recorded and noted so that a word is not tested more than once. The obtained results show that 95.3% of the sub-words or PAW were correctly segmented and



(a) (b) (c)

(d) (e) (f)

Figure 3.9: Successful segmentation of the sub-words using proposed method. All hidden and occluded sub-words are detected

extracted.

Figure 3.9 shows the operation of the proposed algorithm. Diacritics have been automatically ignored by ignoring its cluster from the graph and therefore other than the diacritics every sub-word or PAW has been detected and segmented. When a sub-word is detected it is marked by a red marker once. All images shown in Figure 3.9 have sub-words that are hidden by larger sub-words or has one sub-word begin before the previous sub-word ends and still all of the sub-words were successfully and automatically detected.

Diacritics were set to be ignored in the proposed algorithm but since the font size cannot be enforced in handwritten text some writers wrote the sub-words in the same size as that of diacritics (i.e. 'zer', 'zabr' etc.), which caused that sub-word to become the member of the lowest cluster in the graph. For this reason, some sub-words were rarely considered as diacritics and were thus ignored. This problem has added to the error rate of the proposed method and is shown in Figure 3.10.

Diacritics were ignored because they do not carry any useful information for writer classification or letter recognition. Furthermore, leaving diacritics usually causes errors with a lot of functions, for example the chain code will produce an error for diacritics since the chain code function cannot be used on single points.

Discussing offline Arabic character recognition, the authors in (Sarfraz, 2005) explained

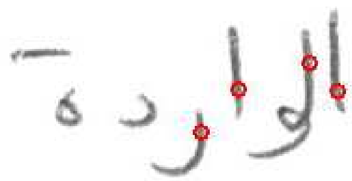


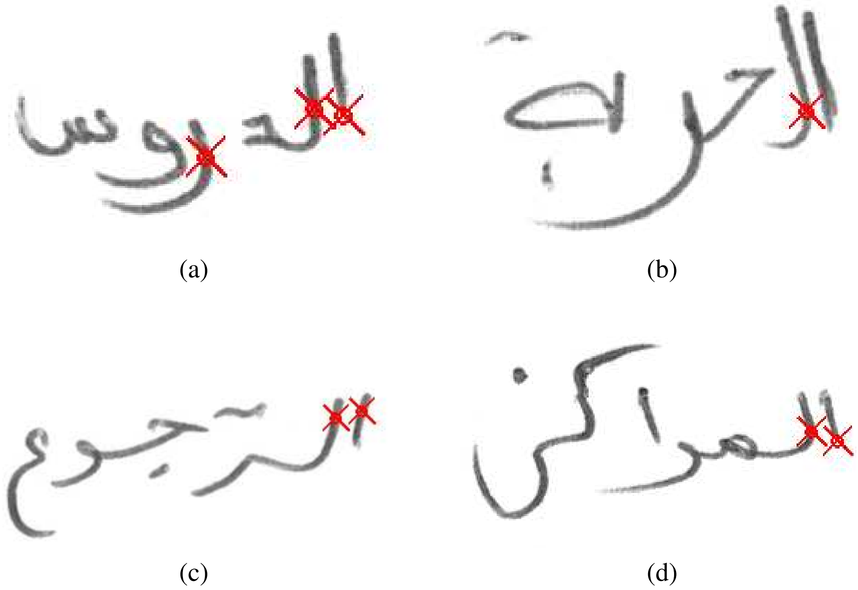
Figure 3.10: Sub-words considered as diacritics. Detected subwords are labeled with red circles

that the characters of Arabic words can be segmented using the projection profile method. The Arabic document was first segmented line by line using the horizontal projection method and then within each line the words were segmented using the vertical projection method. Finally the sub-words were extracted using a combination of baseline detection and vertical projection.

(Lawgali *et al.*, 2011) proposed an algorithm for segmenting words in sub-words and then sub-words in to characters. The proposed algorithm relied heavily on the horizontal and vertical projection method to break up words into sub-words. During the process a large number of over lapping characters were lost and the author considered the lost over lapping characters as noise and had no option but to remove them from the final result.

Similarly, (Osman, 2013) proposed an algorithm for the automatic segmentation of Arabic handwritten text using contour analysis. The author used horizontal projection on an image to find the beginning and end points of the lines. Then, vertical projection was used on each line to find the beginning and end points of each word and sub-word. Due to overlapping a large number of characters were lost and a lot of sub-words were incorrectly segmented. The author also mentioned this loss of characters as a known problem and attributed it to one the reasons that affected the proposed methods result.

For the sake of comparison, the same 537 randomly selected words were tested using the projection profile method used in (Lawgali *et al.*, 2011; Osman, 2013) and it was found that only 79.6% of the sub-words were correctly segmented. The reason for this was that a large number of sub-words were lost due to the way the projection profile method works, that is, the sub-words were either hidden by the bigger sub-words or one sub-word started before the previous one finished. The results of the sub-word segmentation based on projection profile method is shown in Figure 3.11.



(a) (b) (c) (d)

Figure 3.11: Segmentation of sub-words based on projection profile method. The hidden or occluded sub-words are not detected here

The same method of displaying the segmentation results is used in Figure 3.11, where every detected sub-word has been highlighted with a red marker. The results of Figure 3.11 show that even when the sub-words clearly appear to be separate from one another but if one starts before the previous finishes (from a vertical projection point of view), it would appear as a single sub-word to the projection method. A lot of information that can be used in the writer classification is lost in this way. This loss of information stands true for other processes as well. For example, if the chain code algorithm is to be used on each detected sub-word in order to find the writing path, it would only be able to trace the detected sub-words (the one with the red marker) while the others would be ignored, as was the case in (Osman, 2013).

To test that the proposed algorithm works even in the presence of noise, samples were taken from the AHTID/MW dataset and were subjected to “salt and pepper” noise. These noisy samples were then tested on the proposed algorithm for segmentation. Figure 3.13 shows that the presence of noise did not affect the results of segmentation.

3.4 Conclusion

In this chapter a novel approach is proposed for the segmentation of sub-words within handwritten Arabic words. The proposed algorithm was tested on the AHTID/MW dataset and

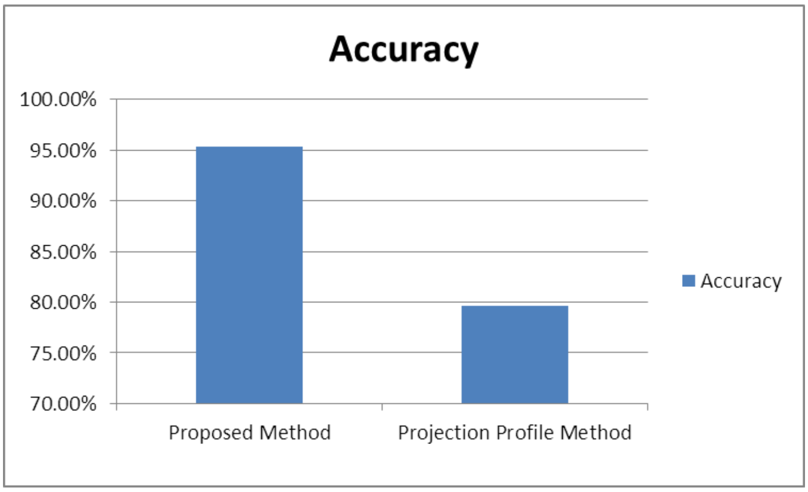


Figure 3.12: Accuracy of the proposed method compared with the projection profile method used previously

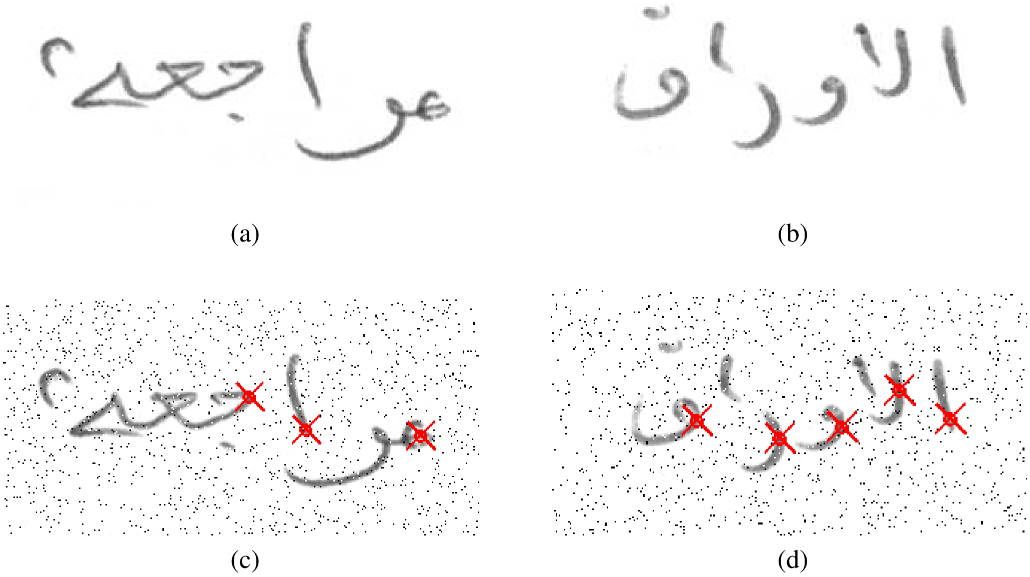


Figure 3.13: Successful segmentation in the presence of noise. (a) and (b) represent the original image, (c) and (d) represent the noisy segmented images

achieved an accuracy of 95.3% in correctly segmenting the sub-words, thus resulting in considerable improvement over the projection profile method. It is worth mentioning that the next chapters will not build upon the method proposed here. The reason for this was due to the fact that this research targeted a script-independent approach rather than focusing on only Arabic script. For script independent writer identification it was determined that extraction of connected components was sufficient and although effective, there is no need for further extraction of sub-words.

The next chapter will discuss an offline text-independent writer identification system using the concept of a multi-scale Local Ternary Pattern Histograms (LTPH) for feature extraction. A classifier is trained for every scale after dimensionality reduction of the features using Spectral Regression Kernel Discriminant Analysis (SR-KDA). Classification is performed by the ensemble of these classifiers.

Chapter 4

Writer Identification using Ensemble of Multi-Scale Local Ternary Pattern Histograms

4.1 Introduction

Over the past few decades, considerable amount of research has been done in the field of writer identification, of which some of the more notable ones have already been discussed in Chapter 2. It can be argued that some of the most effective methods of writer identification can be grouped into either a local approach or global approach. With local approaches the system relies heavily on the segmentation of text before extracting specific features and as is the case with any pattern recognition system, if segmentation fails the system as a whole is compromised. In order to avoid such problems, global approach for writer identification was adopted. In global approaches the identification is based on the general look and feel of the handwriting. Thus, by considering the handwriting as a texture, segmentation can be avoided.

Tan et al. (Tan & Triggs, 2010) proposed the Local Ternary Pattern (LTP) for face recognition. This algorithm was an extension of the Local Binary Patterns (LBP), initially proposed by (Ojala *et al.*, 1996) and later improved in (Ojala *et al.*, 2002). LTP was introduced to overcome the drawbacks and limitations of the LBP i.e. sensitivity to noise and distortions. Later, LTP had been applied successfully in many real world applications including writer identification (Hannad *et al.*, 2016). However, the limitation of the traditional LTP is that it works at only a single resolution and therefore may not always detect the most dominant features.

In this chapter, we propose a multi-scale local ternary pattern histogram feature (MLTPH)

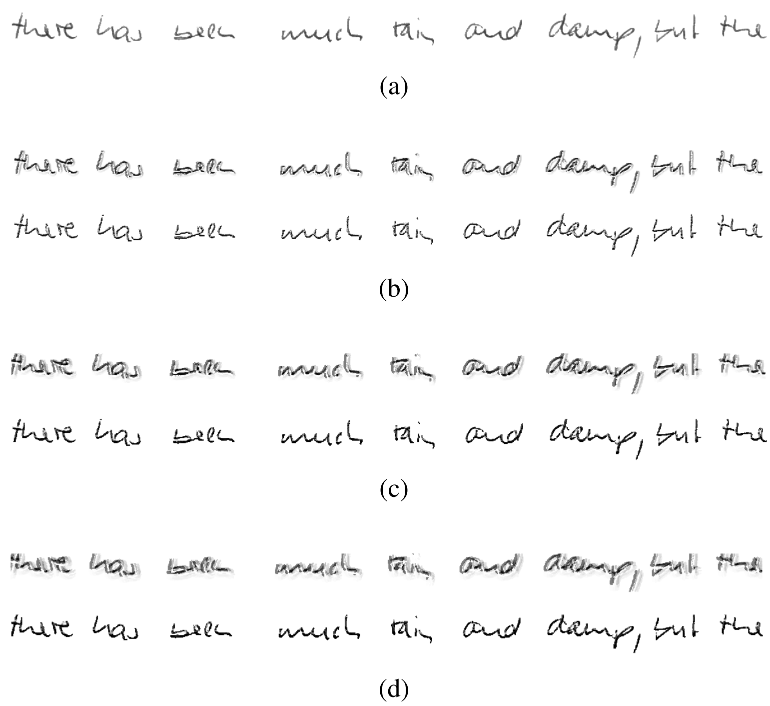


Figure 4.1: (a) Original image. (b) LTP at $R = 3$. (c) LTP at $R = 6$. (d) LTP at $R = 9$.

to improve the performance of the writer identification task (Khan *et al.*, 2016). The LTP features are extracted at various scales in order to best capture the most dominant features of every writer. Figure 4.1 shows the original image and the positive and negative LTP patterns (explained in Section 4.2.1) of the original image at three different scales. As can be seen from the images, the shape and edge information of the text becomes more apparent at multi-scale and thus more dominant features can be extracted at multi-scale rather than single scale.

A general problem that one is usually faced with when using multi-scale feature extractors is the high dimensionality of the data. This produces problems such as the curse of dimensionality and may cause over-fitting of the data. In order to avoid such problems, linear as well as non linear dimensionality reduction techniques were used and compared. The linear methods involved Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) whereas the non-linear methods involved Kernel PCA, Diffusion maps, Local Linear Embedding (LLE), Hessian LLE and Kernel Discriminant Analysis using Spectral Regression (SR-KDA). Of all the methods compared, SR-KDA (Cai *et al.*, 2007) gave the best results in terms of accuracy. The proposed system has been evaluated on two writer identification datasets; IAM (English) and AHTID/MW (Arabic). It has also been compared with the state of the art techniques in the field of writer identification and the results achieved show significant improvement in the Top 1 accuracy when compared with recent techniques.

The remainder of this chapter is organized as follows. Section 4.2 presents the proposed MLTPH approach. Experiments and results are presented in Section 4.3, and finally the chapter is concluded in Section 4.4.

4.2 Proposed System

In this section, we present our proposed system of a multi-scale local ternary pattern histogram. The framework for multi-resolution analysis has been developed by the signal processing, biological vision and computer vision communities and is based on the observation that in the real world all objects are composed of structures at different resolutions or scales. The same can be considered true for writing styles as well.

4.2.1 Multi-Scale Local Ternary Pattern Histograms

The LTP (Tan & Triggs, 2010), shown in Equation 4.1, extracts a local greyscale invariant 3-valued code (1,0,-1) of the image. LTP makes use of a threshold t, in order to encode the difference between the central pixel i_c, and its neighbouring pixels, i_n. The threshold, t along with the central pixel, i_c are used to calculate a tolerance interval for that patch of image. The output of the LTP code is 0 if the pixel intensity is between this interval, any values that are above this interval are set to 1 and any values below this interval are set to -1. For example, for $i_c = 34$ and $t = 5$ the tolerance interval would be [29,35]. Values of the neighbouring pixels that fall within this interval would be set to 0, values greater than 39 would be set to 1 and values below 29 would be set to -1 as shown in Figure 4.2.

$$LTP_{N,R}(x,y) = \sum_{n=0}^{N-1} s'(i_n, i_c, t)2^N \tag{4.1}$$

where N represents the neighbourhood, R represents the scale at which LTP is extracted, t represents a user specified threshold and s' is represented as:

$$s'(i_n, i_c, t) = \begin{cases} 1 & i_n \geq i_c + t \\ 0 & i_c - t \leq i_n < i_c + t \\ -1 & otherwise \end{cases} \tag{4.2}$$

These equations extend the binary pattern of LBP to a ternary LTP pattern. The user specified threshold allows the generated code to be more resistant to noise and distortions. For simplicity the authors in (Tan & Triggs, 2010) suggested another variation of the LTP

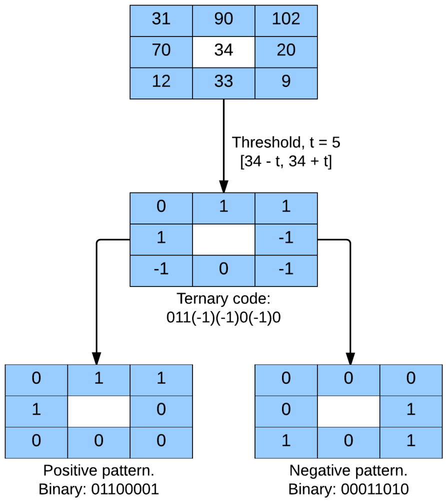


Figure 4.2: An example of the LTP encoding procedure for a 3×3 block.

code where the ternary code is divided into positive and negative patterns. This separates a single code into two channels of LBP descriptors for which separate histograms can be computed. The LTP encoding procedure of a 3×3 block and its splitting into positive and negative patterns is shown in Figure 4.2.

As the name implies, the Multi-scale Local Ternary Pattern Histogram (MLTPH) extends the traditional LTP to a multi resolution representation. This is achieved by applying the traditional LTP operator shown in Equation 4.1 at various radii, R. A multi resolution representation has already been shown to be more effective than a single resolution representation (Chan *et al.*, 2007). However, as explained in Section 4.1 this multi-level representation comes at the cost of high dimensionality and in order to avoid the problems associated with high dimensionality each histogram is subjected to dimensionality reduction via SRKDA (Cai *et al.*, 2007).

4.2.2 Spectral Regression Kernel Discriminant Analysis (SR-KDA)

Linear Discriminant Analysis (LDA) can be considered as one of the most popular methods for dimensionality reduction (Cai *et al.*, 2007) and Kernel Discriminant Analysis (KDA) is the non-linear version of LDA. LDA can be extended to non-linear by considering a kernel matrix, K of size $n \times n$ which is generated from the training vectors. Let $x_j \in \mathscr{R}^d, j = 1, \cdots, k$ represent the training data and $K(x_a, x_b) = \langle \wp(x_a), \wp(x_b) \rangle$. Here, $\wp(x_a)$ $\wp(x_b)$ represent the embedding of x_a and x_b. Now let ρ represent a projection function into the kernel space, then the KDA objective function is represented as:

$$\max_{\rho} D(\rho) = \frac{\rho^T S_b \rho}{\rho^T S_t \rho} \tag{4.3}$$

where S_t represents the total scatter matrix in the feature space and S_b represents the between class scatter matrix in the feature space. It was proved in (Baudat & Anouar, 2000) that Equation 4.3 can be considered equivalent to:

$$\max_{\sigma} D(\sigma) = \frac{\sigma^T KBK\sigma}{\sigma^T KK\sigma} \tag{4.4}$$

where $\sigma = [\sigma_1, \sigma_2, \ldots.., \sigma_n]^T$ is an eigenvector that satisfies $KBK\sigma = \lambda KK\sigma$. $B = (B_j)_{j=1,\ldots.n}$ is a $(n \times n)$ diagonal block matrix of writer labels arranged such that the upper half represent the positive labels whereas the lower half represents the negative labels. Every eigenvector σ gives a projection function ρ into the feature space.

It was later shown in (Cai *et al.*, 2011) that rather than solving the KDA eigenprobelm, the KDA projections can be calculated using the following two equations

$$\begin{aligned} B\psi &= \lambda\psi \\ (K + \delta I)\sigma &= \psi \end{aligned} \tag{4.5}$$

where ψ is the eigenvector of B, I represents the identity matrix and $\delta > 0$ represents a regularization parameter. Gram-Schmidt method is used in order to obtain eigenvectors ψ. Since $(K + \delta I)$ is positive definite, the linear equations in 4.5 can be solved using Cholesky Decomposition

$$K^* \sigma = \psi \Leftrightarrow \begin{cases} U^T \beta = \psi \\ U\sigma = \beta \end{cases} \tag{4.6}$$

i.e. first the vector β must be determined and then the vector σ. U represents an upper triangular matrix, such that $U^T \times U = K^*$.

Once σ is determined the test sample data are calculated as: $f(x) = \sum_{i=1}^{n} \sigma_i K(x, x_i)$ where $K(x, x_i) = \langle \wp(x), \wp(x_i) \rangle$ and prediction can be performed using the projected data.

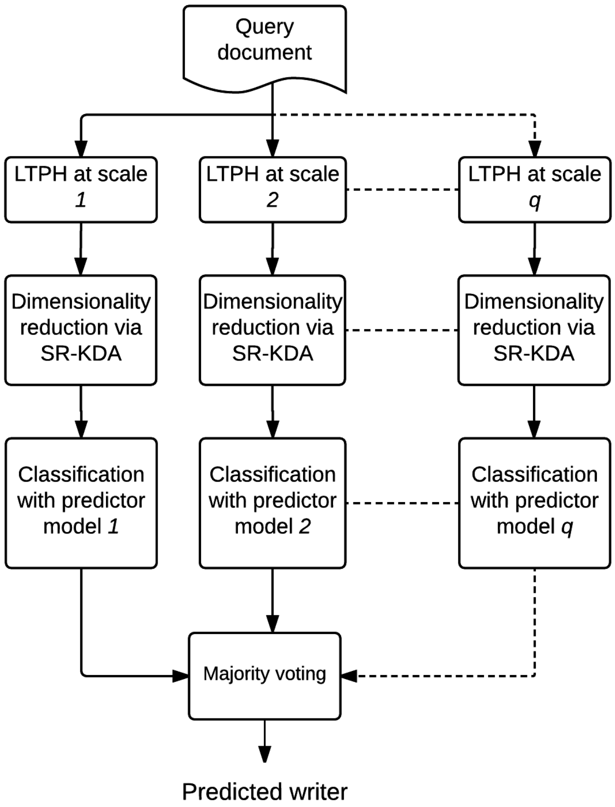


Figure 4.3: Classification of a query document via majority voting.

4.2.3 Ensemble of Classifiers using Majority Voting Rule

Let the feature vectors of writer W_1 extracted at a given scale be represented by L_1 such that $L_1 = \{(a_m, b_m), m = 1 \dots P\}$, where a represents the matrix of feature vectors, b represents the class labels of those feature vectors and m represents the number of samples. From the feature vectors extracted at this scale a predictor model $\phi_1(a, L_1)$ can be generated and then used to predict the label of an unknown query image. Similarly, another model $\phi_2(a, L_2)$ can be generated for the same writer W_1 from the feature vectors extracted at the next scale, L_2. In this way, q predictor models can be generated from L_q training sets $\{\phi_q(a, L_q)\}$. During classification process, an unknown query image is compared to all of these models and their predictions recorded. The final decision is made by aggregating the predictions of all the models using majority voting (Kittler *et al.*, 1998).

The process of classification of a query document via majority voting is shown in Figure 4.3. LTPH features are first extracted from the query document at q different scales. These features are then subjected to dimensionality reduction via SR-KDA and finally compared with their respective predictor models. The predictions from every model are then aggregated via majority voting. The writer having the majority votes from all the predictor models is considered as the most probable author of the query document.

4.3 Experiments and Results

The proposed system is applied on two challenging datasets; IAM and AHTID/MW. For the sake of comparable test conditions, the IAM dataset is used as was arranged by the authors of the dataset themselves. The authors suggested a 100 writer identification task which has a well-defined training, test and validation sets. The AHTID/MW dataset is arranged so that $2/3$ of the dataset is used for training while the remaining is used for testing. The final result displayed here is obtained after a 4-fold cross validation so that it is comparable with other works using the same dataset.

The images from all the writers are subjected to LTP encoding at 12 different scales or levels of resolution i.e. $R = [1,2,3\dots12]$ and the results are recorded at every level. It was observed that no single resolution worked best for every writer. Some writers identified better under one scale whereas others performed better under a different scale. To overcome this problem, every writer is subjected to LTP encoding at every level of R. The histograms obtained are subjected to dimensionality reduction and finally identification is performed via nearest neighbour classifier. Classifying the images at every level of R results in 12 models (one for every scale). Similarly, the query or test documents also undergo the same process and are compared against every model. The results obtained are subjected to majority voting and the identified writer having the majority vote from all the 12 models is selected as the identified writer. Figure 4.4 shows the comparison of the Top 1 accuracy achieved on the IAM dataset against every resolution of R. As can be seen, the best results are achieved when the results from all the models are combined and subjected to majority voting.

A comparison of the proposed system with the state of the art systems in the field of writer identification for the IAM and AHTID/MW dataset are shown in Table 4.1 and Table 4.2 respectively.

It is clear that the proposed system outperforms the other systems in both datasets. Top 1 accuracies of 99.4% and 87.5% were achieved using the IAM and AHTID/MW datasets, respectively. There is an improvement of 1% for the IAM dataset when compared to the nearest best approach by (Schlapbach & Bunke, 2006) that uses the same number of writers. Similarly, for the AHTID/MW dataset an improvement of 10.2% was achieved when compared

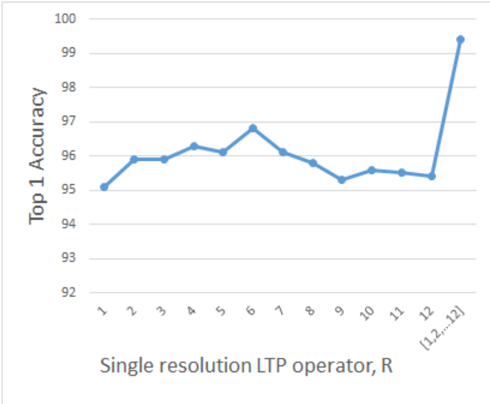


Figure 4.4: Comparison of the Top 1 accuracy achieved on the IAM dataset with different values of R.

System	Number of Writers	Top 1 Accuracy
Schlapbach and Bunke (Schlapbach & Bunke, 2004)	50	96.5%
Bensefia et al. (Bensefia *et al.*, 2005b)	150	86.0%
Kırlı et al. (Kırlı & Gülmezoğlu, 2012)	93	98.7%
Schlapbach and Bunke (Schlapbach & Bunke, 2007)	100	97.0%
Schlapbach and Bunke (Schlapbach & Bunke, 2006)	100	98.4%
Khalifa et al. (Khalifa *et al.*, 2015)	100	98.0 %
Schomaker and Bulacu Schomaker & Bulacu (2004)	100	81.2%
Schomaker et al. Schomaker *et al.* (2004)	100	60.4%
Proposed system	100	**99.4%**

Table 4.1: Comparison of the proposed system with the state of the art systems for the IAM dataset.

System	Number of Writers	Top 1 Accuracy
Slimane and Margner (Slimane & Margner, 2014)	53	69.4%
Schomaker and Bulacu Schomaker & Bulacu (2004)	53	46.2%
Schomaker et al. Schomaker *et al.* (2004)	53	34.8%
Hannad et al. Hannad *et al.* (2016)	53	77.3%
Khalifa et al. (Khalifa *et al.*, 2015)	53	44.3%
Proposed system	53	**87.5%**

Table 4.2: Comparison of the proposed system with the state of the art systems for the AHTID/MW dataset.

to the nearest best performing approach by (Hannad *et al.*, 2016). The performance of our proposed system on the Arabic AHTID/MW dataset is noteworthy. It was observed by Bulacu et al. (Bulacu *et al.*, 2007) that the writer identification systems developed primarily for Latin scripts performed very poorly when applied to Arabic handwriting. It was concluded that writer identification on Arabic handwriting is more challenging when compared with other languages such as Roman and Latin. The results achieved by our system demonstrate that more than acceptable results have been achieved for both languages and that the challenging Arabic script does not result in our system to suffer greatly.

4.4 Conclusion

In this chapter, a segmentation free multi-scale local ternary pattern histogram-based system for writer identification has been proposed. LTP histograms are calculated at various scales. A writer identification model is generated for every level and the query document is compared against every model of every writer. The final decision is based on the majority voting rule i.e. the prediction from every model is recorded and the writer that has the majority votes from the various scales is considered the most probable author of the query document. The proposed system has been applied on an Arabic and an English dataset and the results achieved show that the presented system outperforms the previous state of the art systems in both languages.

The next chapter will discuss a robust offline text-independent writer identification algorithm using bagged Discrete Cosine Transform (DCT) features. DCT features are extracted at a local level using an overlapping sliding window. Our proposed Bagged DCT (BDCT) approach allows us to utilize the power of the DCT (i.e. robustness to noise and distortions) while avoiding the problems associated with it. SRKDA has been used for dimensionality reduction and finally, classification is done using nearest centre rule and majority voting.

Chapter 5

Robust Off-line Text Independent Writer Identification Using Bagged Discrete Cosine Transform Features

5.1 Introduction

Writer identification using texture based approaches are known to produce good results as they do not depend on segmentation of text. We show in this chapter that when extracting textural features from small overlapping blocks, segmentation can be beneficial in speeding up the overall process and remove white blocks that do not contain any information.

In this chapter, a robust Bagged Discrete Cosine Transform (BDCT) technique is proposed for writer identification (Khan *et al.*, 2017). Discrete Cosine Transform (DCT) has been adopted for a number of reasons: (i) the representation of an image in the DCT domain has been shown to be effective for the purpose of image matching (Mitrea *et al.*, 2004), (ii) its compressive nature as the DCT can represent a block of handwritten text with fewer coefficients while still maintaining most of the handwriting information and (iii) the DCT coefficients are robust to distortions that may occur during the writing or scanning process (noise, blurring, change in contrast, etc.). Figure 5.1 shows two reconstructed blocks of size 64×64 with only the first 2500 coefficients in a zigzag order. As can be seen, the handwriting information can be perfectly recovered.

The extraction of DCT features at such a local level prevents them from being used due to memory limitations. To overcome this problem, random unique features are selected from every image for codebook generation. This universal codebook is then used for generating

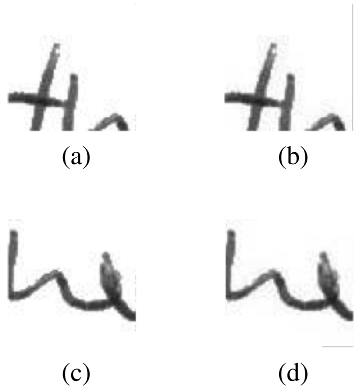


Figure 5.1: Reconstructed blocks with fewer DCT coefficients. (a) original block. (b) reconstruction of (a) with 2500 DCT coefficients. (c) original block. (d) reconstruction of (c) with 2500 coefficients.

descriptors for every image. The BDCT approach allows for DCT features to be effectively exploited for robust hand writer identification. The proposed system has been assessed on the original version of hand written documents of various datasets and the results have shown comparable performance with state-of-the-art systems. Then, blurry and noisy documents of two different datasets have been considered through intensive experiments where the system has been shown to perform significantly better than its competitors. To the best of our knowledge this is the first work that addresses the robustness aspect in automatic hand writer identification. This is particularly suitable in digital forensics as the documents acquired by the analyst may not be in ideal conditions.

In summary, the main components of our proposed system include local descriptor computation using Discrete Cosine Transform, multiple vector quantisation using bagging and clustering, structured writer representation via localised histograms of vector codes, dimensionality reduction using kernel discriminant analysis and classification using nearest centre rule. The proposed system has been evaluated on four hand written datasets including IAM, CVL, AHTID/MW and IFN/ENIT. The results achieved show that the system delivers comparable performance with state-of-the-art systems in the case where query documents are presented in ideal conditions on one hand. On the other hand, the system exhibits robustness against noise and blur unlike existing systems.

The main contributions of this work are twofold: (i) A new BDCT approach is proposed for writer identification which avoids the problems associated with traditional DCT-based feature extraction techniques, i.e., memory limitations due to the abundance of features and undesirable similarity of local features among various writers. (ii) Unlike previous automatic writer identification systems, the proposed system is only marginally affected by distortion and noise; this is mainly due to the DCT feature extractor which is known to be very robust to noise and blurring distortions. The robustness of writer identification is vital in forensic applications where the query data is collected under severe imaging conditions.

The organization of this chapter is as follows: Section 5.2 describes the proposed system using the BDCT concept and also discusses the dimensionality reduction and classification using nearest centre rule. Section 5.3 provides an experimental evaluation of the proposed system, Section 5.4 discusses the results achieved while Section 5.5 is dedicated to the conclusions drawn from our work.

5.2 Proposed System

Feature extraction is accomplished in the DCT domain with the help of an overlapping sliding window. The DCT can be viewed as a projection of the signal onto an orthogonal basis composed of cosine functions. In addition to being a de-correlating transform, the DCT has been widely used in image compression due to its compressive nature (Sayood, 2012). The DCT transforms a block of pixels b of size $N_1 \times N_2$ into a matrix of real numbers as

$$B(u,v) = \frac{2}{N_1 N_2} C(u)C(v) \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \cos\left(\frac{u\pi}{N_1}(i+0.5)\right) \cos\left(\frac{v\pi}{N_2}(j+0.5)\right) b(i,j) \quad (5.1)$$

where $0 \leq u \leq N_1 - 1$ and $0 \leq v \leq N_2 - 1$. $C(0) = \frac{1}{\sqrt{2}}$ and $C(\delta) = 1$ for $\delta \neq 0$.

Each image generates tens of thousands of feature vectors since the image is divided into small overlapping blocks. Such a large amount of data demands a lot of resources in terms of available memory and also may cause over fitting of data. Due to these issues, it is simply not possible to build a model by using traditional DCT. To overcome this problem, random unique features are selected from every image from all writers for the generation of every predictor model. This collection of random features is clustered using a clustering algorithm. Previously three main clustering algorithms have been used for codebook generation i.e. k-means, 1D SOM (Self Organizing Map) and 2D SOM. However it was demonstrated by (Bulacu & Schomaker, 2005) that the clustering method used to generate the codebook did not affect the end result since basically the same performance was observed for all three clustering methods. We have used k-means for clustering of features. The clustering of these select random features from all writers allows for the creation of a universal codebook. In other words, a feature vector of each sample can be generated by producing an occurrence histogram whose bins (equal to the number of centroids used during clustering) correspond to the indices of each feature to its nearest centroid. This histogram of occurrences is then normalized to get the final feature vector. The process of universal codebook generation is shown in Figure 5.2.

Once the universal codebook for every model has been generated, the system can be trained by extracting the descriptor for every handwritten image with respect to its own codebook. These generated feature vectors are of high dimensionality, it is therefore sensible to reduce the dimension of the feature space. Furthermore, the universal codebook generated from the DCT features suffers from the same problem as with many other local feature extractors relating to a high intra-class variance with a long tail distribution. In order to solve this problem, kernel discriminant analysis with spectral regression(SR-KDA) is deployed for reducing the dimensionality of the feature space while decreasing the intra-class variance (Explained in Section 5.2.1). The SR-KDA method creates a predictor model which can be used to identify the writer from a query image. The training phase of the system is shown in Figure 5.3.

It is worth noting that each universal codebook is generated using a set of randomly selected feature vectors from all writers. However, a random selection of features may not always best represent a class and thus may lead to poor classification results. To overcome this problem bootstrap aggregation or bagging is used. Bagging is considered to be one of the most popular re-sampling ensemble methods. The concept of bootstrap aggregating was proposed by (Breiman, 1996) and is based on the assumption that by using multiple versions of a predictor rather than just one, a better result can be achieved by aggregating the results of those predictors.

Let's assume that a learning set L consists of data $\{(y_n, x_n), n = 1 \dots N\}$, where x is the data matrix, y represents the class labels of that data matrix and N represents the number of samples. From this data a predictor model, $\phi(x, L)$ can be generated and the label y of an unknown image can be predicted using this model ϕ. The same learning set L can be divided into a sequence of learning sets $\{L_k\}$ each consisting of N independent observations from the same master learning set L. These k learning sets can be used to generate k predictors, $\{\phi(x, L_k)\}$. In this case each model will predict its own class label y for an unknown image. The final prediction is achieved by aggregating all of the individual predictions by method of either majority voting (Kittler *et al.*, 1998), mean or product (Tao *et al.*, 2006).

The main concept of bagging is that the predictor models generated from the k learning sets will disagree at times due to the variance of the learning sets but this variance is compensated via aggregation. In the proposed approach, the variance of the learning set is obtained by the random selection of DCT features for universal codebook generation. This concept has shown

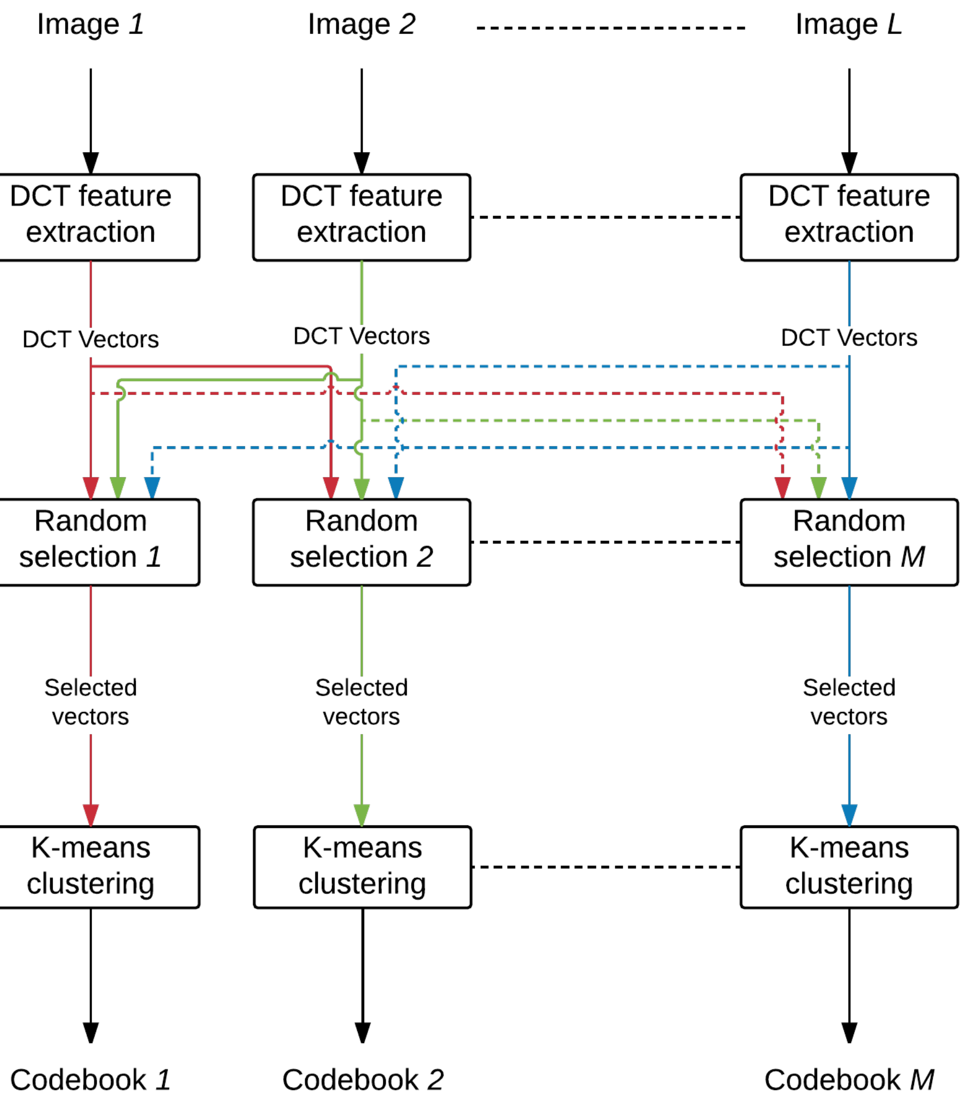


Figure 5.2: Multiple codebook generation with random feature selection. L = number of images, M = number of models, $(L >> M)$

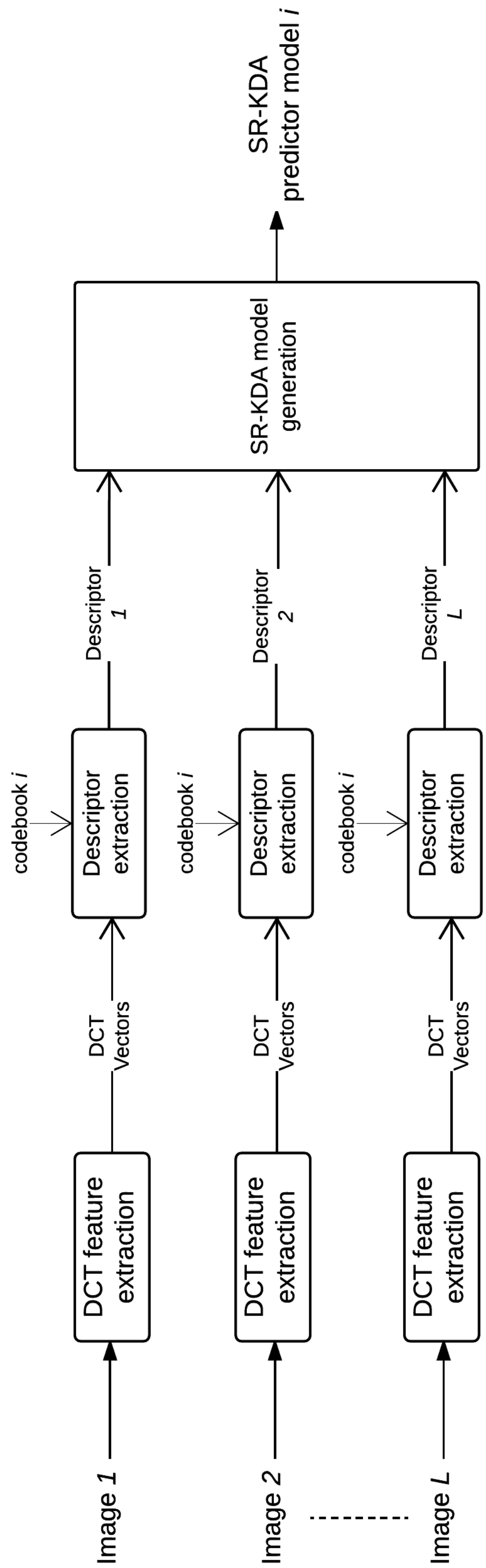


Figure 5.3: Training phase - SR-KDA predictor model i is generated for codebook i.

to provide better results than just using a base model.

The testing phase of our system is shown in Figure 5.4. When an unknown image is presented to the system, its DCT features are extracted in the same manner as in the training phase. The vectors are then used to generate a descriptor (i.e. a histogram) for the test image via each codebook. That is, M descriptors are obtained for each query image. Finally, each descriptor is classified by the corresponding SR-KDA predictor model. The predicted writers from all models are subjected to majority voting and the writer having the majority from all the predictor models is selected as the most probable writer for that unknown test image.

5.2.1 Kernel Discriminant Analysis with Spectral Regression (SR-KDA) for Dimensionality Reduction

Kernel discriminant analysis is a non-linear technique of Linear Discriminant Analysis (LDA) (Cai *et al.*, 2011; Fukunaga, 2013). In LDA, the projection vectors are acquired by decreasing the variation of the same class and at the same time, increasing the between class scatter. Equation 5.2 described the main goal of LDA

$$e_{opt} = argmax \frac{e^T C_b e}{e_T C_t e} \tag{5.2}$$

where C_t and C_b indicate the within and between class scatter matrix respectively. The eigenvectors related to the non-zero eigenvalues of matrix $C_t^{-1} C_b$ are the optimal $e's$.

To extend LDA as non linear, consider kernel matrix K of size $n \times n$ which is computed from the training vectors obtained using codebook generation. Let $x_j \in \mathscr{R}^d, j = 1, \cdots, k$ are the vectors of training data and $K(x_a, x_b) = \langle \wp(x_a), \wp(x_b) \rangle$. Here, $\wp(x_a)$ $\wp(x_b)$ are the embeddings of x_a and x_b. Let us represent the projection function as ρ into the kernel space. Equation 5.3 described the objective function of KDA

$$\max_{\rho} D(\rho) = \frac{\rho^T S_b \rho}{\rho^T S_t \rho} \tag{5.3}$$

where S_t and S_b represent the total and between class scatter matrices respectively in the feature space. Eigen-problem $C_b \rho = \lambda C_t \rho$ which is equivalent to Equation 5.4 as proved by (Baudat & Anouar, 2000) is then used to solve Equation 5.3.

$$\max_{\sigma} D(\sigma) = \frac{\sigma^T KBK\sigma}{\sigma^T KK\sigma} \tag{5.4}$$

where $\sigma = [\sigma_1, \sigma_2, ..., \sigma_n]^T$ is an eigenvector conform to $KBK\sigma = \lambda KK\sigma$. Every eigen-

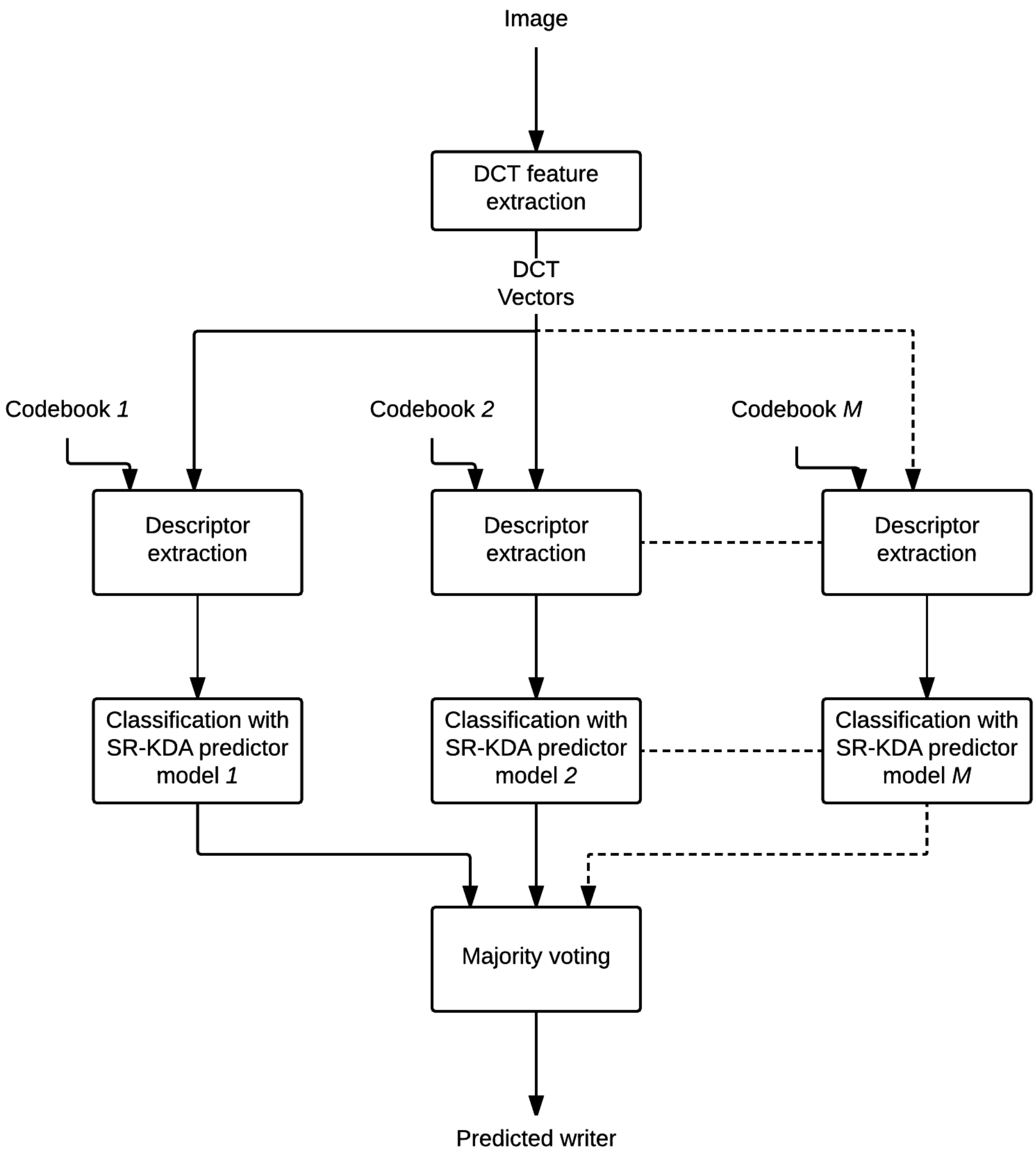


Figure 5.4: Testing phase of the proposed system.

vector σ assigns a projection function ρ into the feature space. $B = (B_j)_{j=1,...,n}$ is a $(n \times n)$ block diagonal matrix of writers or classes.

It is shown in (Cai *et al.*, 2011; Tahir *et al.*, 2015) that the following two linear equations can be used to obtain the KDA projections

$$\begin{aligned} B\psi &= \lambda\psi \\ (K+\delta I)\sigma &= \psi \end{aligned} \tag{5.5}$$

where $\delta > 0$ is a regularization parameter, I is the identity matrix, and ψ is an eigenvector of B. Gram-Schmidt method is utilized to obtain Eigen-vectors ψ. Since $(K+\delta I)$ is positive definite, linear equations in 5.5 are solved using Cholesky Decomposition as follows

$$K^*\sigma = \psi \Leftrightarrow \begin{cases} R^T\beta = \psi \\ R\sigma = \beta \end{cases} \tag{5.6}$$

which initially involves finding vector β and then solving for vector σ. Briefly,

- SR-KDA prevents the computation of eigenvector by solving regularized regression problems.

- The main advantage is its capability to handle large kernel matrices and in the significant reduction of the computational cost. The two main steps when computing SR-KDA are the response generation using Gram-Schmidt method and the use of Cholesky decomposition to solve $(c-1)$ linear equation where c is the number of writers or classes in the training data. Let "flam" be an operation consisting of one multiplication and one addition (Stewart, 1998). $(mc^2 - \frac{1}{3}c^3)$ flams is the total cost of Gram-Schmidt method (Cai *et al.*, 2011) and m^2c flams are required to solve $c-1$ linear equations. The Cholesky decomposition needs $\frac{1}{6}m^3$ flams. Thus, the total cost of SR-KDA is $\frac{1}{6}m^3 + m^2c + mc^2 - \frac{1}{3}c^3$. This cost can easily be approximated as $\frac{1}{6}m^3 + m^2c$. If we compare this cost with ordinary KDA ($\frac{9}{2}m^3 + m^2c$), SR-KDA considerably reduces the most expensive eigenvector computation. It achieves 27 times speed-up over traditional KDA.

- After obtaining σ, test data samples are calculated from : $f(x) = \sum_{i=1}^{n} \sigma_i K(x, x_i)$ where $K(x, x_i) = \langle \wp(x), \wp(x_i) \rangle$ and the projected data can be used for prediction. In this study, nearest centroid classifier (NCC) to get the final decision from each model of SR-KDA (Cai *et al.*, 2011).

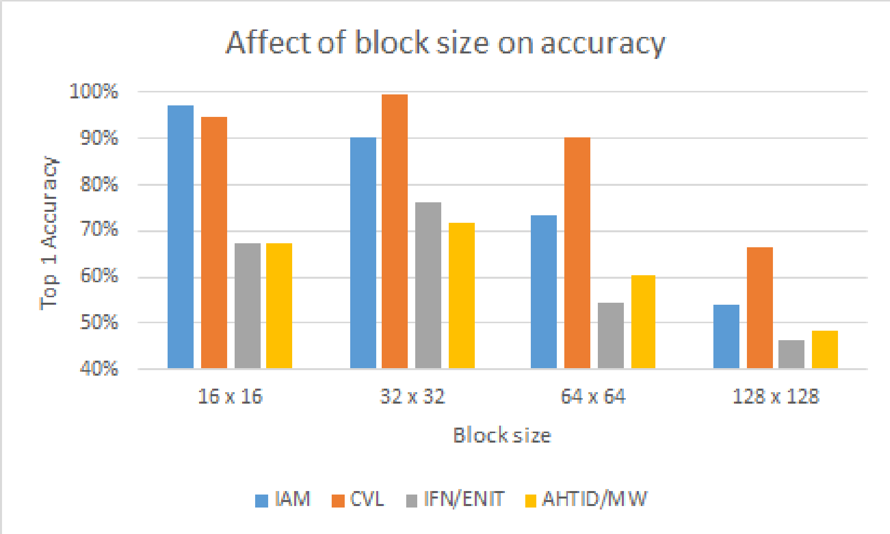


Figure 5.5: Comparison of Top 1 accuracy based on block size.

5.3 Experiments and Results

5.3.1 Extraction of DCT features

Since the DCT is a frequency transform that characterises the significance of changes across adjacent pixels, it is more sensible to use digital documents in the form of grey level images rather than binary images in order to capture as much frequency information as possible. Moreover, to ensure that we do not get an overwhelming return of blocks containing only white spaces the images are first segmented to return all the connected components of that image with respect to a set threshold. The thresholding ensures that the object such as diacritics and accidental ink traces can be ignored. These connected components are then divided into overlapping sliding blocks of size $n \times n$. The block size should be large enough to contain an acceptable amount of information about the writer and also small enough to ensure acceptable identification (Séropian, 2003). The optimum block size was determined empirically and the results achieved with different block sizes are shown in Figure 5.5.

It can be seen that 32×32 block size produced the best results for the CVL, IFN/ENIT and AHTID/MW datasets whereas a block size of 16×16 produced the best results for the IAM dataset. For each block, DCT features were extracted and saved using a zig-zag scan pattern as described by (Robinson & Kecman, 2003). This zig-zag extraction (shown in Figure 5.6) allows for converting the 2-D DCT matrix into a 1-D vector of size 1024 (for our 32 x 32 block size). The magnitude of the coefficients decreases as we travel down the vector and this allows us to use less coefficients (if need be) while still maintaining the most important part of the block.

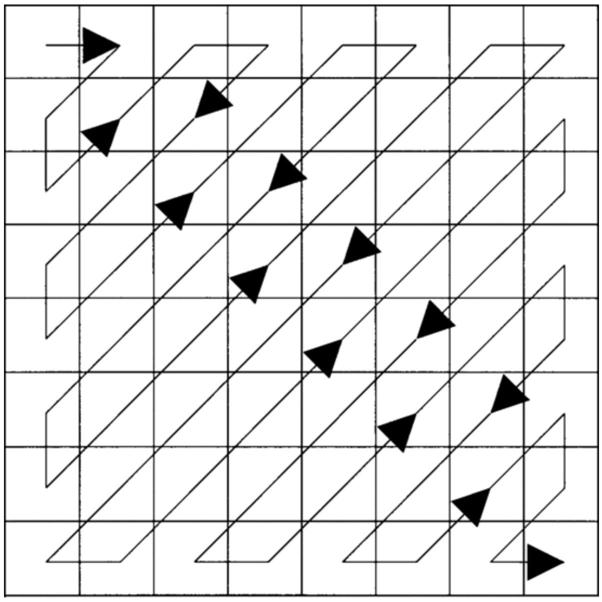
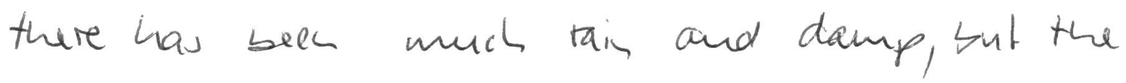


Figure 5.6: Zig-zag extraction of the DCT coefficients.

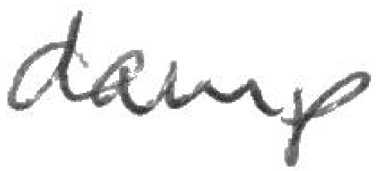
This process of dividing an input image into connected components and then further dividing these components into blocks of size $n \times n$ is shown in Figure 5.7.

Recall from Section 5.2 that a random selection of DCT feature vectors is performed to create a number of codebooks. The codebook size (i.e. the number of centroids used in clustering) affects the accuracy achieved and the optimum codebook size for our system was determined through experimentation. The codebook size should be sufficiently large to represent the variability in the feature space but on the other hand it should not cause over fitting. The different codebook sizes tested and their accuracy achieved is shown in Figure 5.8. The codebook size of 1500 proved to be the best for the CVL, for the Arabic AHTID/MW and IFN/ENIT datasets a codebook size of 2000 worked best whereas for the IAM dataset best results were achieved with a codebook size of 500. IAM dataset performing better with a smaller codebook size can be related to the small amount of data available per writer since by using the modified version of the dataset each writer is left with a very limited amount of text. Note that the descriptors (i.e. histogram vectors) of all writers obtained using a codebook are subjected to dimensionality reduction via SR-KDA.

The main improvement in our proposed system lies in the generation of multiple SR-KDA predictor models for every writer which in turn were generated from universal codebooks of random selection of DCT features. Since our system is based on a majority voting rule from all predictor models to generate the final result it would only make sense that the higher the



(a)



(b)

(c)

Figure 5.7: (a)An image sample from the IAM dataset (b) One of the words extracted from the line (c) The word divided into overlapping blocks of size 32 x 32.

number of models the more consistent the result would be. Since every predictor depends on randomly selected features, there exist models which are generated using features that may not completely represent the writer. Therefore, although more models would theoretically produce better results there must exist a point beyond which using more predictors do not bring any significant gain. This level needed to be determined as generating a model is computationally expensive. Our proposed system was tested with models, $\phi = 5, 7, 10, 12, 14, 16, 18, 20, 22, 25, 30$ to determine the optimal number of models needed. Figures 5.9, 5.10, 5.11 and 5.12 show the accuracy achieved from the majority voting of the various models generated for the IAM, CVL, AHTID/MW and the IFN/ENIT datasets respectively. A steady increase in accuracy can be observed up until 20 models. After which the results show that further increasing the models after 20 has no significant effect on the performance of the overall system.

5.3.2 Comparison of our proposed system with existing work

A comparative study was performed in order to compare the performance of the proposed system with the state of the art techniques already published in the field of writer identification. As discussed earlier, experiments were performed on the IAM, CVL, AHTID/MW and IFN/ENIT datasets. The arrangement of these datasets have been explained in Section 2.2.

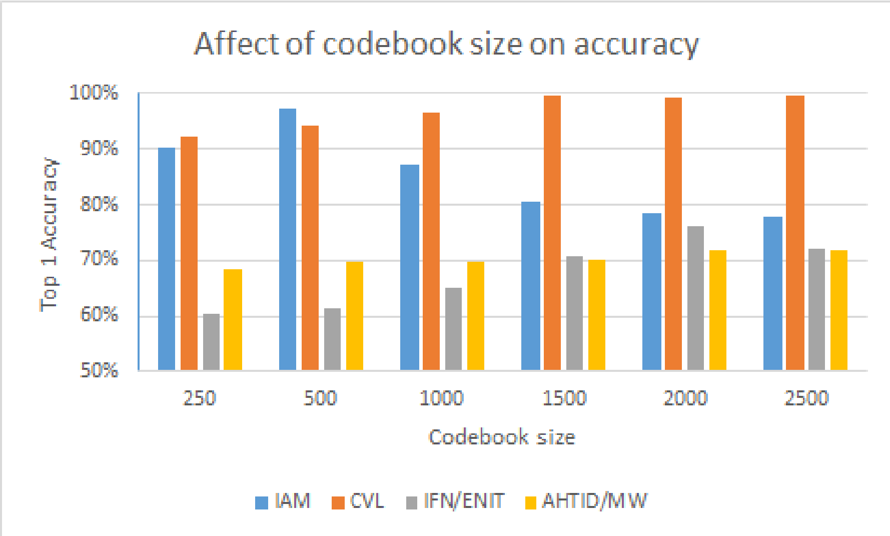


Figure 5.8: Comparison of the Top 1 accuracy based on codebook size.

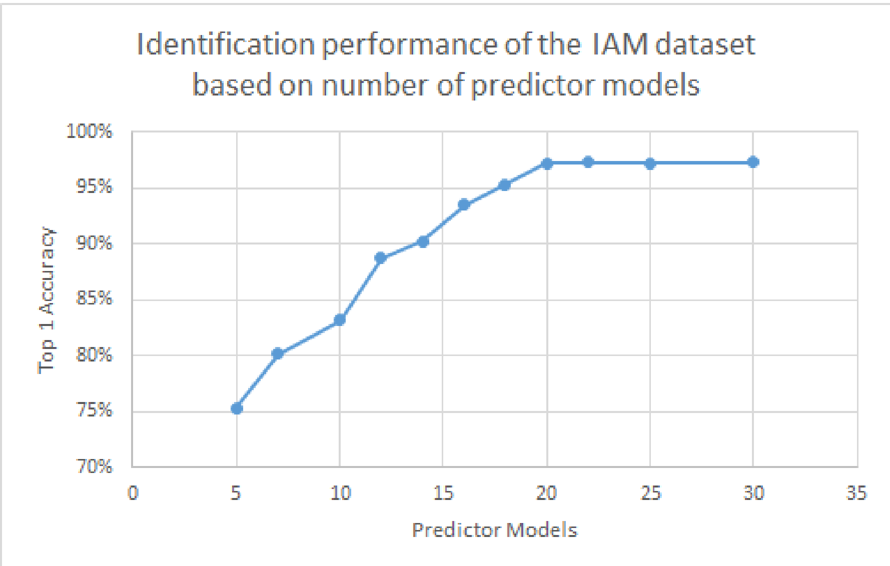


Figure 5.9: Comparison of the Top 1 accuracy of the IAM dataset based on number of predictor models.

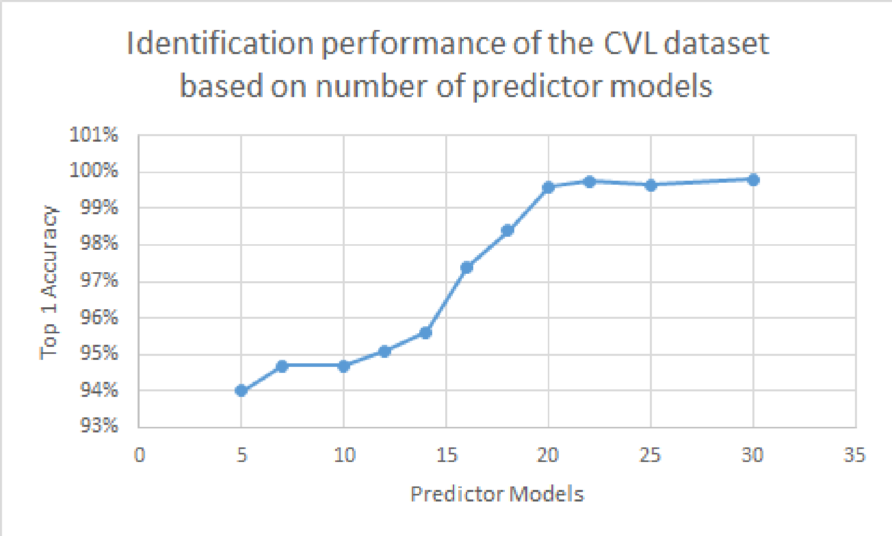


Figure 5.10: Comparison of the Top 1 accuracy of the CVL dataset based on number of predictor models.

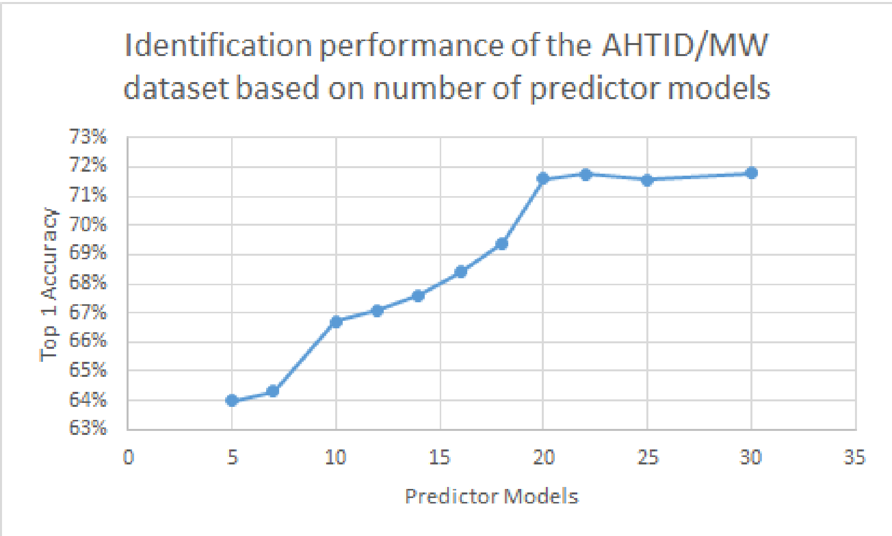


Figure 5.11: Comparison of the Top 1 accuracy of the AHTID/MW dataset based on number of predictor models.

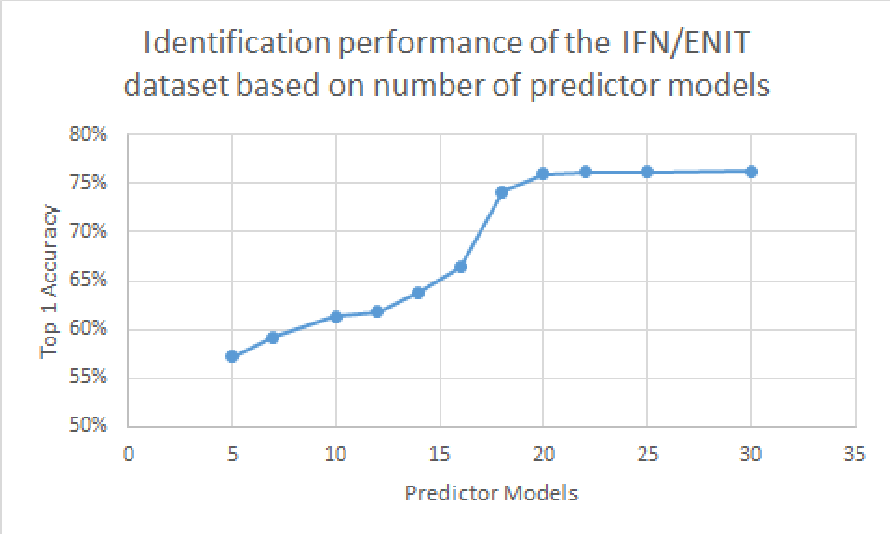


Figure 5.12: Comparison of the Top 1 accuracy of the IFN/ENIT dataset based on number of predictor models.

Using our proposed BDCT approach, a Top 1 accuracy of 97.2% on the IAM dataset has been achieved, which outperforms the nearest best performing system of (Bertolini *et al.*, 2013) by 0.5%. For the CVL dataset, 99.6% of Top 1 accuracy has been reached by our system. This outperforms by 0.2% the nearest best system developed by (Jain & Doermann, 2014). For the AHTID/MW dataset, 71.6% of Top 1 accuracy has been obtained with the proposed system which is still comparable to the state of the art, outperformed only by (Khan *et al.*, 2016) and (Hannad *et al.*, 2016). For the IFN/ENIT dataset, however, the system shows a clear drop in performance. Note that the images of this dataset are given in binary format and the system seems to be severely affected by this type of images when compared to existing techniques. This was expected since the DCT features describe the frequency content of images (see Section 5.3.1). In fact, because binary images carry extremely little frequency information, the documents written by different writers would have similar frequency contents if they were represented in binary form, i.e., the inter-class similarity increases drastically due to binarization.

5.3.3 Robustness of the proposed system

In practice, handwritten samples under investigation are not always presented to the forensic analyst in ideal conditions. The samples could be noisy or blurry due to the imaging conditions under which they have been collected. It is imperative that identification algorithms are robust enough to ignore such distortions.

System	Number of writers	Top 1 Accuracy
Bulacu & Schomaker (2007)	650	89.0%
Siddiqi & Vincent (2010)	650	91.0%
Kumar *et al.* (2014)	650	88.4%
Ghiasi & Safabakhsh (2013)	650	93.7%
Bertolini *et al.* (2013)	650	96.7%
Khalifa *et al.* (2015)	650	92.0%
(Jain & Doermann, 2014)	657	94.7%
Hannad *et al.* (2016)	657	89.5%
Schomaker & Bulacu (2004)	657	82.5 %
Proposed system	**650**	**97.2%**

Table 5.1: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the IAM dataset.

System	Number of writers	Top 1 Accuracy
(Fiel & Sablatnig, 2013)	309	97.8%
(Jain & Doermann, 2014)	310	99.4%
(Christlein *et al.*, 2014)	310	99.2%
(Fiel & Sablatnig, 2015)	309	98.9%
Schomaker & Bulacu (2004)	310	81.8%
Hannad *et al.* (2016)	310	96.2%
Proposed system	**310**	**99.6%**

Table 5.2: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the CVL dataset.

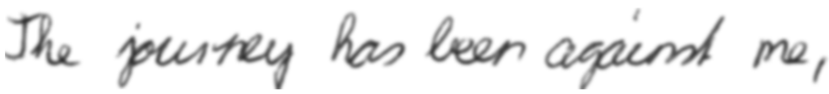
System	Number of writers	Top 1 Accuracy
(Slimane & Margner, 2014)	53	69.4%
Schomaker & Bulacu (2004)	53	46.2%
Schomaker *et al.* (2004)	53	34.8%
Hannad *et al.* (2016)	53	77.3%
(Khalifa *et al.*, 2015)	53	44.3%
(Khan *et al.*, 2016)	**53**	**87.5%**
Proposed system	53	71.6%

Table 5.3: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the AHTID/MW dataset.

System	Number of writers	Top 1 Accuracy
Bulacu & Schomaker (2007)	350	88.0%
(Chawki & Labiba, 2010)	130	82.0%
(Djeddi & Souici-Meslati, 2011)	130	84.2%
(Abdi & Khemakhem, 2012)	100	85.0%
Abdi & Khemakhem (2015)	411	90.0%
Hannad *et al.* (2016)	**411**	**94.9%**
Proposed system	411	76.0%

Table 5.4: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the IFN/ENIT dataset.

(a)



(b)

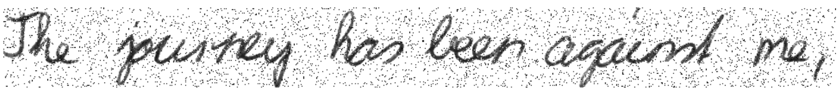
(c)

(d)

(e)

Figure 5.13: Gaussian blurring applied to a sample of text from the IAM dataset. (a) Original image. (b) Gaussian filter with standard deviation of 2. (c) Gaussian filter with standard deviation of 3. (d) Gaussian filter with standard deviation of 4. (e) Gaussian filter with standard deviation of 5.

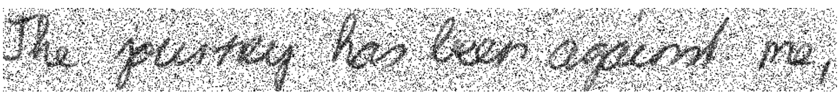
(a)



(b)

(c)

(d)



(e)

Figure 5.14: Salt & pepper noise applied to a sample of text from the IAM dataset. (a) Original image. (b) Salt & pepper with a noise density of 0.1. (c) Salt & pepper with a noise density of 0.2. (d) Salt & pepper with a noise density of 0.25. (e) Salt & pepper with a noise density of 0.3.

System	dataset Used	Accuracy Reported	Our Implementation
Schomaker & Bulacu (2004)	Firemaker	94.0%	92.3%
Hannad *et al.* (2016)	IAM	89.5%	88.7%

Table 5.5: Comparison of reported accuracy of published works against our implementation of the same.

To demonstrate the robustness of the proposed system, the AHTID/MW dataset and 100 randomly selected writers from the IAM dataset were subjected to two types of distortion; blurring with a low pass Gaussian filter and "salt & pepper" noise. This noise was applied at incrementally increasing levels. The application of blurring and "salt & pepper" noise to samples of the IAM dataset can be seen in Figure 5.13 and Figure 5.14, respectively. These noisy versions of the dataset were used to record the Top 1 accuracy of the proposed system along with two other systems previously published in literature as proposed by (Schomaker & Bulacu, 2004) and (Hannad *et al.*, 2016). Furthermore, a variation of our proposed system was also applied on the noisy datasets, where SIFT was used for feature extraction instead of DCT. SIFT is the preferred feature extractor for purposes related to object detection in images which has also been widely used for the purpose of writer identification (Fiel & Sablatnig, 2012; Fiel & Sablatnig, 2013; Wu *et al.*, 2014; Xiong *et al.*, 2015). To verify our implementation of the systems used in our experimental comparison, i.e. (Schomaker & Bulacu, 2004) and (Hannad *et al.*, 2016), the same datasets adopted in the original papers have been used with similar settings. For example, (Schomaker & Bulacu, 2004) applied their system on the Firemaker dataset (Schomaker & Vuurpijl, 2000). Only the uppercase handwriting samples from 150 different writers were considered in their study. A codebook was generated from the samples of 100 writers while the samples of another set of 150 writers were used for evaluation by splitting each document in half. The top half of each full document was used as the reference document whereas the bottom half was used as the query document. In (Hannad *et al.*, 2016), the authors applied their system on the full IAM dataset while retaining a maximum of 14 text lines per writer. For each writer 60% of the text lines were used as a reference while the remaining 40% were used for evaluation. The results are depicted in Table 5.5.

As can be seen, the identification results obtained are very close to those reported in the original works. The minor difference, which is less than 2%, is probably due to some tiny variations in experiments which are beyond our control such as differences in segmented lines/connected components and the writers and/or paragraphs used for training and testing purposes.

For a system which is robust against image distortions, the identification results achieved

System	Standard Deviation				
	1	2	3	4	5
Proposed system	**1.2%**	**2.4%**	**2.5%**	**2.7%**	**3.1%**
Our implementation of Hannad *et al.* (2016)	1.1%	2.2%	5.4%	6.5%	7.6%
Our implementation of Schomaker & Bulacu (2004)	15.3%	23.3%	37.8%	56.4%	57.6%
Proposed system with SIFT	1.9%	3.6%	6.2%	9.6%	13.7%

Table 5.6: Drop in Top 1 accuracy observed for the IAM dataset subjected to Gaussian blurring when compared to the results achieved with the noiseless version of the dataset.

System	Noise Density				
	0.05	0.1	0.2	0.25	0.3
Proposed system	**0.2%**	**0.7%**	**2.7%**	**4.3%**	**10.1%**
Our implementation of Hannad *et al.* (2016)	38.1%	62.0%	75.2%	84.3%	89.0%
Our implementation of (Schomaker & Bulacu, 2004)	45.8%	62.9%	86.8%	89.7%	90.0%
Proposed system with SIFT	4.5%	5.8%	10.5%	17.1%	21.1%

Table 5.7: Drop in Top 1 accuracy observed for the IAM dataset subjected to salt & pepper noise when compared to the results achieved with the noiseless version of the dataset.

on the distorted query documents must not differ significantly from those achieved on original documents. For this reason, noise was applied at incrementally increasing levels to the IAM and AHTID/MW datasets and the drop in performance was observed. Table 5.6 and Figure 5.15 show the drop in performance on the IAM dataset for blurring when compared to the noiseless results. Table 5.7 and Figure 5.16 show the drop in performance on the IAM dataset for salt & pepper noise when compared to the noiseless results. Likewise, Table 5.8 and Figure 5.17 show the decrease in performance on the AHTID/MW dataset, where the blurring operation is considered on query documents, when compared against the results obtained on original documents. The drop in performance is also illustrated by Table 5.9 and Figure 5.18 on the AHTID/MW dataset where the query documents are affected by the salt & pepper noise. As can be seen, the proposed system shows a slight decrease in performance for all the tested noisy and blurry documents, whereas the competing systems suffer from massive performance drops and can no longer operate effectively in such conditions.

It can also be seen that that the proposed system exhibits robustness against both types of distortion of various intensities and outperforms the competing systems including the SIFT-based variation of our system. This also illustrates the efficiency and suitability of DCT features for forensic applications. The “salt & pepper” noise proved to be a more challenging task for all the systems but at every level of intensity the proposed system achieved more than

System	Standard Deviation				
	1	2	3	4	5
Proposed system	**1.8%**	**4.2%**	**6.1%**	**7.5%**	**9.5%**
Our implementation of Hannad *et al.* (2016)	2.5%	4.9%	9.7%	14.6%	17.1%
Our implementation of (Schomaker & Bulacu, 2004)	3.5%	6.3%	12.0%	43.2%	48.9%
Proposed system with SIFT	2.5%	5.2%	8.3%	16.5%	22.0%

Table 5.8: Drop in Top 1 accuracy observed for the AHTID/MW dataset subjected to Gaussian blurring when compared to the results achieved with the noiseless version of the dataset.

System	Noise Density				
	0.05	0.1	0.2	0.25	0.3
Proposed system	**7.7%**	**10.5%**	**17.6%**	**22.3%**	**25.7%**
Our implementation of Hannad *et al.* (2016)	47.2%	51.5%	63.3%	74.4%	91.7%
Our implementation of (Schomaker & Bulacu, 2004)	43.2%	67.9%	77.3%	85.2%	94.7%
Proposed system with SIFT	4.1%	11.8%	31.1%	40.4%	55.3%

Table 5.9: Drop in Top 1 accuracy observed for the AHTID/MW dataset subjected to salt & pepper noise when compared to the results achieved with the noiseless version of the dataset.

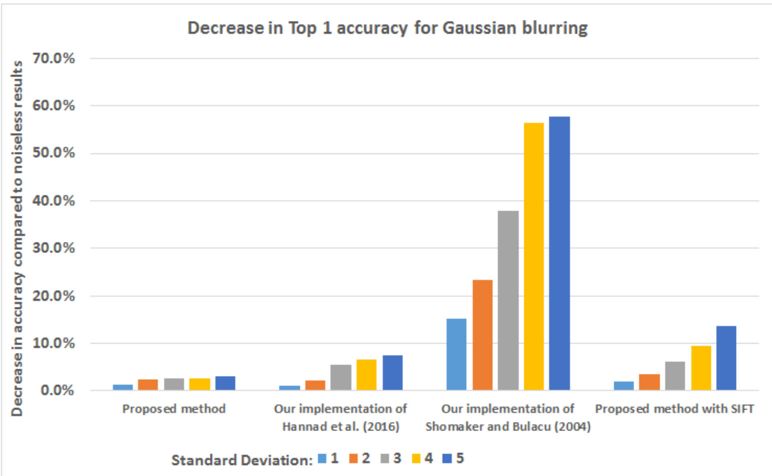


Figure 5.15: Comparison of the drop in accuracy observed for the IAM dataset subjected to Gaussian blurring with incrementally increasing standard deviation.

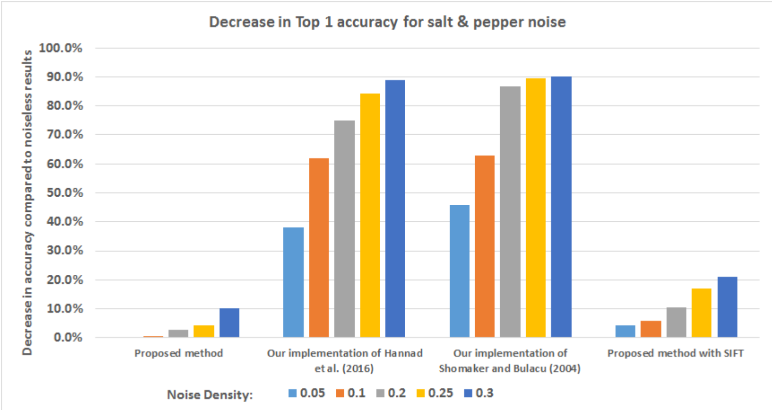


Figure 5.16: Comparison of the drop in accuracy observed for the IAM dataset subjected to salt & pepper noise with incrementally increasing noise density.

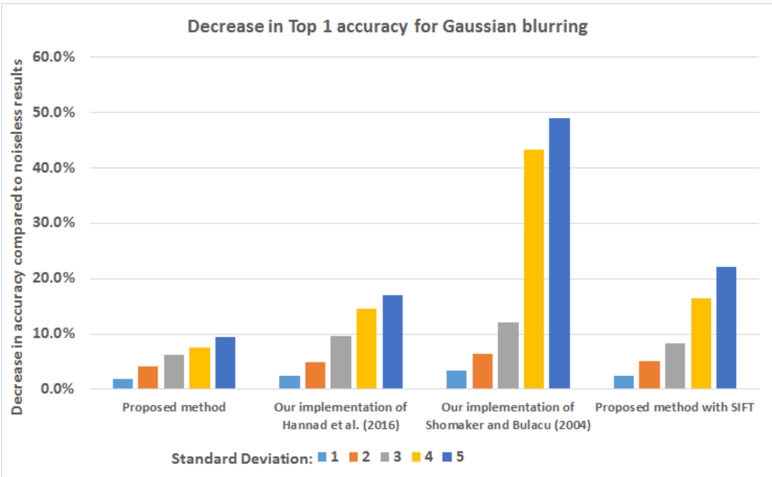


Figure 5.17: Comparison of the drop in accuracy observed for the AHTID/MW dataset subjected to Gaussian blurring with incrementally increasing standard deviation.

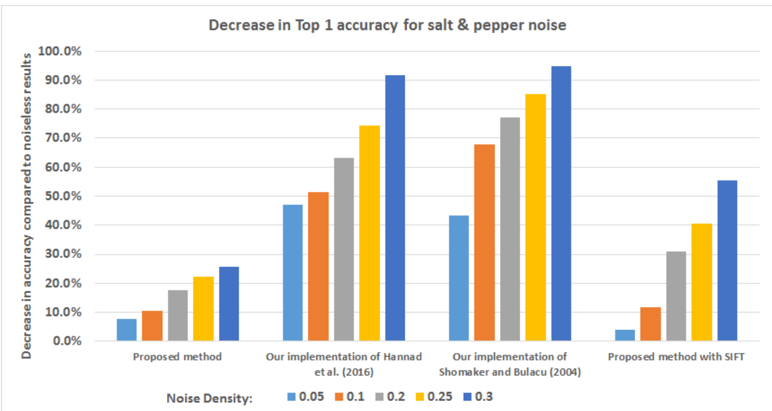


Figure 5.18: Comparison of the drop in accuracy observed for the AHTID/MW dataset subjected to salt & pepper noise with incrementally increasing noise density.

acceptable results.

5.4 Discussion

The proposed BDCT approach compares well with state-of-the-art hand writing identification systems when the query documents are presented at a reasonably good visual quality. However, while existing systems fail to maintain acceptable performance when the query documents are subjected to noise and blurring, the proposed BDCT system shows significant improvements. Robustness is one of the main strengths of the proposed system. However, as mentioned earlier, the nature of the features used, i.e., DCT-based, suggest that the system cannot perform well on documents presented in binary form. Furthermore, by analysing the results obtained for all four databases, it is clear that the Arabic script results are not as good as those of the Latin script counterpart. This suggests that the identification of Arabic scripts is more challenging due to the high similarity that, sometimes, exists across different characters on one hand, and the small flexibility of writing the same character in different patterns on the other hand.

5.5 Conclusion

In this chapter, a robust system for offline text independent writer identification has been proposed using the concept of universal codebooks with bagged DCT features. Multiple SR-KDA predictor models have been generated for each writer and a majority voting rule is used to make the final decision on an unknown query document. Our proposed BDCT approach allows DCT features, that have been extracted from overlapping blocks, to be effectively used for automatic hand writer identification. It also allows us to avoid the problems associated with DCT features extracted at such a small scale i.e. memory limitations due to abundance of features and similar local features among various writers. The proposed system exploits the robustness property of the DCT features in hand writer identification. Experiments performed on noisy and blurry versions of query documents taken from two different datasets demonstrate a clear superiority of the proposed system over state-of-the-art techniques in noisy and blurry conditions. Furthermore, by analysing the results obtained for all four datasets, it is clear that the Arabic script results are not as good as those of the Latin script counterpart. This suggests that the identification of Arabic scripts is more challenging due to the high similarity that, sometimes, exists across different characters on one hand, and the small flexibility of writing the same character in different patterns on the other hand.

The next chapter presents an efficient handwriting identification system that combines SIFT and RootSIFT descriptors in a set of Gaussian mixture models (GMM). In particular, a new concept of similarity and dissimilarity Gaussian mixture models (SGMM and DGMM) is introduced. A new weighted histogram method is also proposed to derive an intermediate prediction score for each writer's GMM.

Chapter 6

New Gaussian Mixture Models for Efficient Offline Handwritten Text-Independent Identification using SIFT and RootSIFT Descriptors

6.1 Introduction

This chapter proposes an offline writer identification system that relies on a similarity and dissimilarity Gaussian mixture model (SGMM and DGMM) approach using a weighted histogram of Gaussian Mixture Model (GMM) scores. Scale Invariant Feature Transform (SIFT) and RootSIFT (Arandjelović & Zisserman, 2012) descriptors, which are used to represent handwritten text data are employed to construct a set of SGMMs and a DGMM for every writer (as will be explained in Section 6.2.3). In this context, for every writer, two SGMMs are generated to describe the intra-class similarity that is exhibited between handwritten texts of a writer using SIFT and RootSIFT descriptors. On the other hand, the DGMM represents the dissimilarity which inherently exists between that writer's style and other different handwritings of other different hand writers styles using SIFT technique. While the SGMM/DGMM approach leads to multiple scores for a single handwritten text generated from key point-based descriptors, a new weighted histogram method is introduced to efficiently derive intermediate prediction scores for each writer's GMM. The proposed system has been evaluated using six publicly available datasets including multiple languages: three English, two Arabic and one Hybrid language. A comparative analysis against state-of-the-art systems has been carried out

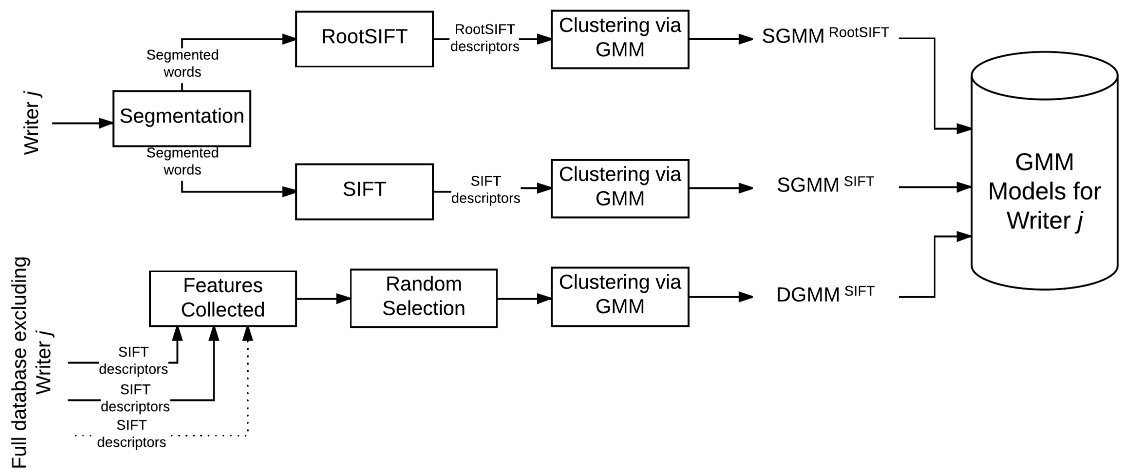


Figure 6.1: SGMM and DGMM generation for writer j.

to validate the proposed approach.

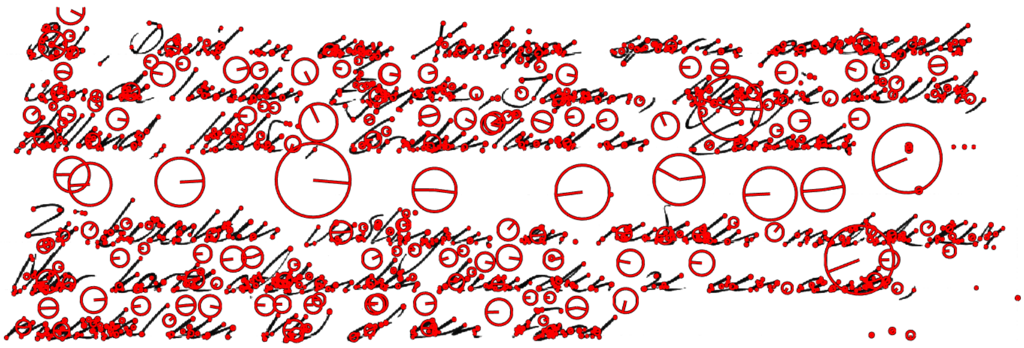
The remainder of this chapter is organised as follows. Section 6.2 describes the proposed method in detail. Section 6.3 provides an experimental analysis of the proposed system along with a discussion of the results achieved. And finally, Section 6.4 discusses the conclusions drawn from this work.

6.2 Proposed System

The framework for the training phase of the proposed system is illustrated by Fig.6.1. It consists of three phases: segmentation of handwritten text, feature extraction and generation of identifier models. A detailed explanation of each phase is given below.

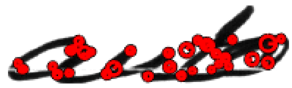
6.2.1 Word Segmentation

For handwriting identification, the features can be extracted from either the allographs within the words, the full words or from the full page. However, for the purpose of handwriting analysis, the features extracted at the word level are much more effective than those extracted at a page or allograph level (Wu *et al.*, 2014). This is because the features extracted from a full page may include other features that are detected between the words and lines, as shown in Fig.6.2(a). These features contain no relevant information and are in fact detrimental to the identification procedure. Allograph level features are extracted at a sub-word level and may lead to many stable identifying features of the writer to be missed. The features extracted at a word level perform well because only strong and valid identifying features are extracted as exemplified by in Fig.6.2.

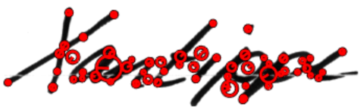


(a)

(b)



(c)



(d)

(e)

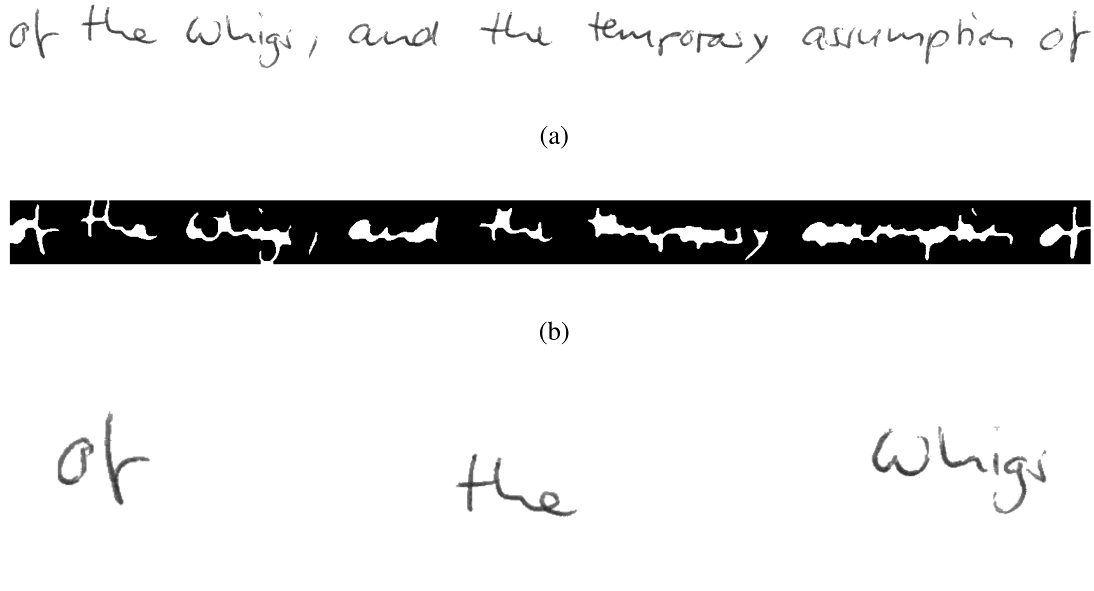
Figure 6.2: (a) SIFT feature extraction on an unsegmented document from the Firemaker dataset showing invalid keypoints. (b) (c) (d) (e) Only valid SIFT keypoints extracted after word segmentation.

The extraction of words from a document is achieved by using a mask on the original grey scale image. To obtain this mask, the input image is subjected to a binarization process using, for example, Otsu's method (Otsu, 1975) which can be represented mathematically as

$$g(x,y) = \begin{cases} 1 & if\, f(x,y) > T \\ 0 & otherwise \end{cases} \tag{6.1}$$

where $f(x,y)$ is the original image, $g(x,y)$ is the output image and T is the threshold at which the image is binarized.

After the binarization process, a morphological closing, which is a process of performing dilation followed by an erosion of the image, is performed on the logical image using a disk structuring element. The structuring element is kept at a certain size to ensure that the gaps between letters of words are closed but those between words are still distinguishable. This ensures that the individual words are extracted and the white gaps between words are ignored. Once this step is done, the extraction of the words can be carried out using a bounding box. The area of each bounding box is then used to determine and remove the diacritics, commas and periods since they do not carry any information related to the writing style of the writer. The bounding box coordinates allows for the extraction of the connected components which are used as masks on the original image to extract only the sample text words while ignoring all the white spaces. This procedure is shown in Fig.6.3.



(a) (b) (c) (d) (e)

Figure 6.3: Segmentation and word extraction procedure. (a) An image from the IAM dataset in it's original form. (b) The same image after being subjected to binarization and morphological closing. (c) (d) (e) The extracted words.

6.2.2 Feature Extraction

SIFT method was first proposed by Lowe in (Lowe, 2004) and has been successfully applied in many fields due to its capability to extract very distinctive and scale invariant features from images (Nister & Stewenius, 2006). SIFT is usually the preferred method of feature extraction in applications such as object retrieval and object detection. In particular, a writer can generate text with varying scale, orientation and translation. Furthermore, the different scanning procedures of the documents can cause variations in the illumination for the training and query documents. Therefore, these challenges should be taken care of by the feature extraction method to ensure robustness against such variations in order to provide a reliable result. SIFT has proven to be an efficient method to address these geometric distortions as demonstrated in the field of writer identification (Fiel & Sablatnig, 2012; Fiel & Sablatnig, 2013; Wu *et al.*, 2014; Xiong *et al.*, 2015). SIFT algorithm operates in four stages. First, an image is broken down into a Gaussian pyramid of octaves where the original image is then convolved with its corresponding octaves of the pyramid with difference of Gaussian (DoG) filters at different variances. In the next stage, referred to as key point localization, the stable key points are detected. Then the orientations, scales and locations of these key points are calculated. Finally, 128 dimension descriptors are generated to represent the image features. This is based on the histogram of oriented gradients (HoG).

In addition to SIFT we have also employed RootSIFT (Arandjelović & Zisserman, 2012) for the purpose of feature extraction. RootSIFT and SIFT follow the same principle for the extraction of the features with the only difference being that SIFT uses an Euclidean distance for similarity measurement while RootSIFT uses the Hellinger kernel. By using the Hellinger kernel instead of the Euclidean one, significant performance improvements can be obtained (Arandjelović & Zisserman, 2012). This is due to the fact that the Euclidean distance is much less efficient than the Hellinger kernel for comparison of histograms.

Let us analyse the connection between the Hellinger kernel and the Euclidean distance kernel in SIFT. Let x and y to be two feature vectors having a unit Euclidean normalization, i.e. $\|x\|_2 = 1$. Therefore, it follows, the relationship between the Euclidean distance, $d_E(x,y)$ and their similarity kernel is given as

$$\begin{aligned} d_E(x,y)^2 &= \|x-y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^T y \\ &= 2 - 2S_E(x,y) \end{aligned} \tag{6.2}$$

where

$$S_E(x,y) = x^T y \text{ and } \|x\|_2^2 = \|y\|_2^2 = 1.$$

To convert SIFT to RootSIFT the Euclidean kernel need be replaced with the Hellinger kernel. The Hellinger kernel is defined as (Nikulin, 2001):

$$H(x,y) = \sum_{i=1}^{n} \sqrt{x_i y_i} \tag{6.3}$$

where x and y represent two L_1-normalized histograms, i.e. $\sum_i^n x_i = 1\,,\, x_i \geq 0$.

Therefore, to perform a similarity measure between two SIFT descriptors using the Hellinger kernel, two algebraic operations must be followed, (i) perform an L_1 normalization of the SIFT descriptor and (ii) perform an element wise square root operation on the normalized SIFT vector. Therefore,

$$S_E(\sqrt{x}, \sqrt{y}) = \sqrt{x}^T \sqrt{y} = \sum_{i=1}^{n} \sqrt{x_i y_i} = H(x,y) \tag{6.4}$$

where

$$S_E(\sqrt{x}, \sqrt{x}) = \sqrt{x}^T \sqrt{x} = \sum_{i}^{n} x_i = 1 \tag{6.5}$$

At this stage, the SIFT descriptors have been converted to RootSIFT and as such, comparing these RootSIFT descriptors using the Euclidean distance will have the same effect as comparing original SIFT vectors via the Hellinger kernel, i.e.

$$d_E(\sqrt{x}, \sqrt{y})^2 = 2 - 2H(x,y) \tag{6.6}$$

By following this procedure, the benefits of using the Hellinger kernel on SIFT descriptors can be exploited without altering the original script used to generate the SIFT vectors. As a result, SIFT can be simply replaced with RootSIFT at every point of the algorithm. It is worth noting that each segmented word in the handwritten text document provides a number of SIFT and RootSIFT descriptors. Therefore, the total number of SIFT and RootSIFT descriptors extracted from each text document varies depending on the number of segmented words as well as the number of key points detected on each word.

6.2.3 Similarity and Dissimilarity Gaussian Mixture Models

GMM's have been widely used and successfully applied in the field of speech recognition (Reynolds, 1995; Reynolds *et al.*, 2000). In this work, a GMM models the distribution of the feature vectors extracted from an individual's handwritten text by a multivariate Gaussian mixture distribution (Reynolds *et al.*, 2000). This model is then used to estimate the probability that a certain handwritten text image corresponds to that individual's handwriting style.

Given a feature vector x with D random variables representing an individual's handwriting style, the multivariate Gaussian mixture density conditioned on a set of parameters λ_j for the j^{th} writer ($j \in \{1,2,\cdots,N\}$) is defined as

$$p(x|\lambda_j) = \sum_{i=1}^{M} \phi_i^j \; \aleph(x|\mu_i^j, C_i^j), \tag{6.7}$$

where $\aleph(x|\mu_i^j, C_i^j)$ stands for the multivariate Gaussian function with mean vector $\mu_i^j \in \Re^{D\times 1}$ and covariance matrix $C_i^j \in \Re^{D\times D}$. ϕ_i^j is the weight corresponding to the i^{th} Gaussian where $\sum_{i=1}^{M} \phi_i^j = 1$. Here, the parameters of the GMM, i.e., $\lambda_j = \{\mu_i^j, C_i^j, \phi_i^j\}$, are estimated via the Expectation Maximization (EM) algorithm (Dempster *et al.*, 1977) in an iterative fashion. Once the parameters λ_j are estimated for writer j, the model can be used to calculate the probability conditioned on λ_j for a query text image where the extracted feature vectors are $X = \{x_1, x_2, \cdots, x_K\}$. Conventionally, the negative log-likelihood can be used to identify a writer by selecting the GMM that corresponds to the lowest value. This is given as

$$-\log p(X|\lambda_j) = \sum_{k=1}^{K} -\log p(x_k|\lambda_j). \tag{6.8}$$

However, because the query feature vectors within the same document might be considerably similar or different from the features used to construct the GMMs, the summation given by (6.8) may not be effective practically since it assigns the same weight to the contribution of descriptors. To address this problem, we propose in this paper a weighted histogram-based method that efficiently derives the intermediate prediction score of the GMM for any given query handwritten text image as will be detailed later.

Let us first discuss the three types of GMM models, proposed for each writer, as illustrated by Fig.6.1. All three models are combined to determine the identity of query documents. The RootSIFT features extracted from the training documents of writer j are used to generate a GMM identifier for that writer. Similarly, another GMM identifier for the same writer is generated from the corresponding SIFT features. These two GMM identifiers can be considered to be the authority on identifying the handwriting of writer j and they are viewed as similarity GMMs (SGMM) because they describe the intra-class features. On the other hand, a Dissimilarity GMM (DGMM) is also trained for the same writer (writer j). The DGMM takes into consideration the contrast between the text written by writer j and the rest of the writers from the dataset. The DGMM is generated from the SIFT features of the full dataset excluding the writer of interest, i.e., writer j. This ensures that the DGMM will cover a good range of the negative population if writer j is taken as a reference.

Note that the DGMM utilizes SIFT features and not the RootSIFT ones. This is because it was observed that the DGMM based on RootSIFT features did not bring any improvements in terms of the identification accuracy, whereas the DGMM generated from the SIFT features greatly helped in improving performance. This can be justified by the fact that due to the process by which SIFT features are converted to RootSIFT (Section 6.2.2) the divergence between the writers is minimized. This decrease in divergence helps in creating an identifier model for a single writer but does not help when one considers the contrast between a writer and the rest of writers in the dataset. To demonstrate this point, Fig.6.4 and Fig.6.5 show the intra-class and inter-class divergence for RootSIFT and SIFT respectively using samples from the AHTID/MW dataset.

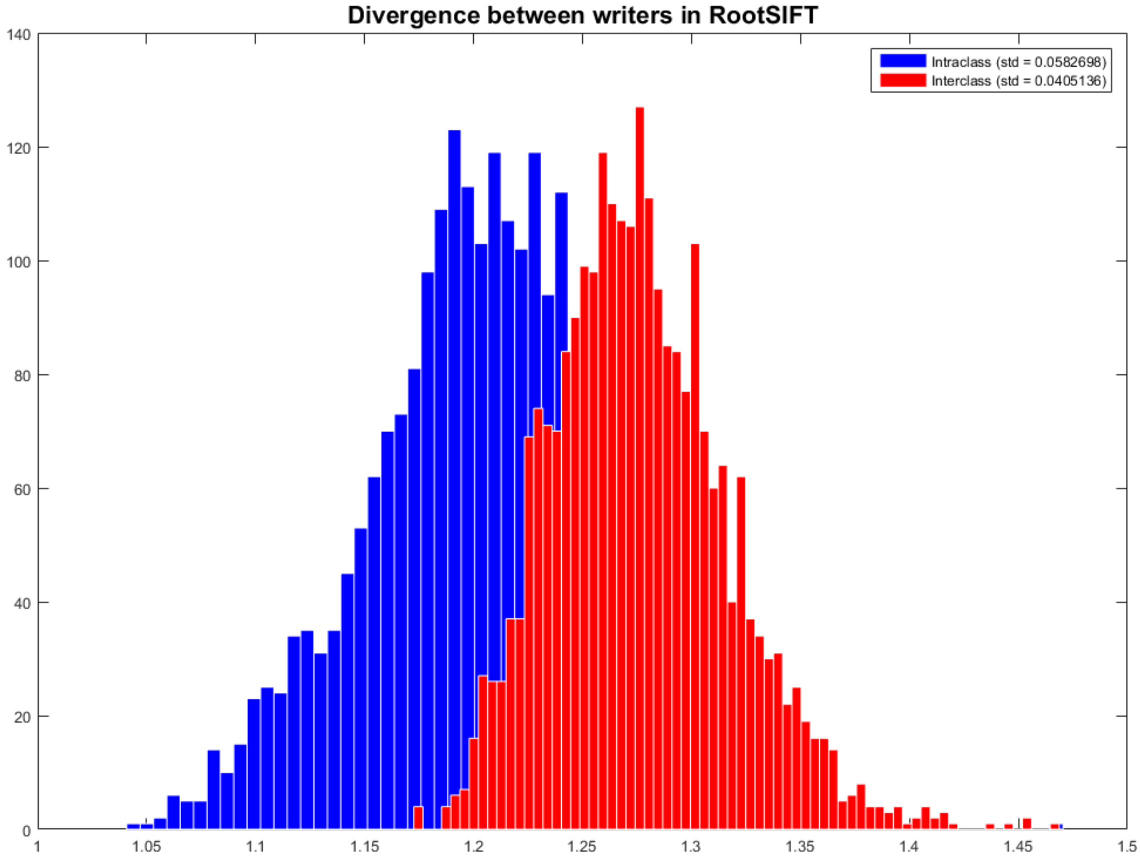


Figure 6.4: Divergence between the writers of the AHTID/MW dataset in RootSIFT

As can be seen for RootSIFT, the divergence between the interclass documents is very small (signified by a standard deviation of 0.04). On the contrary, the divergence between the inter-class documents for SIFT is noticeably high (shown by a standard deviation of 9.69). Therefore, the DGMM appears to contribute in identifying the query document using SIFT but fails to do so with RootSIFT.

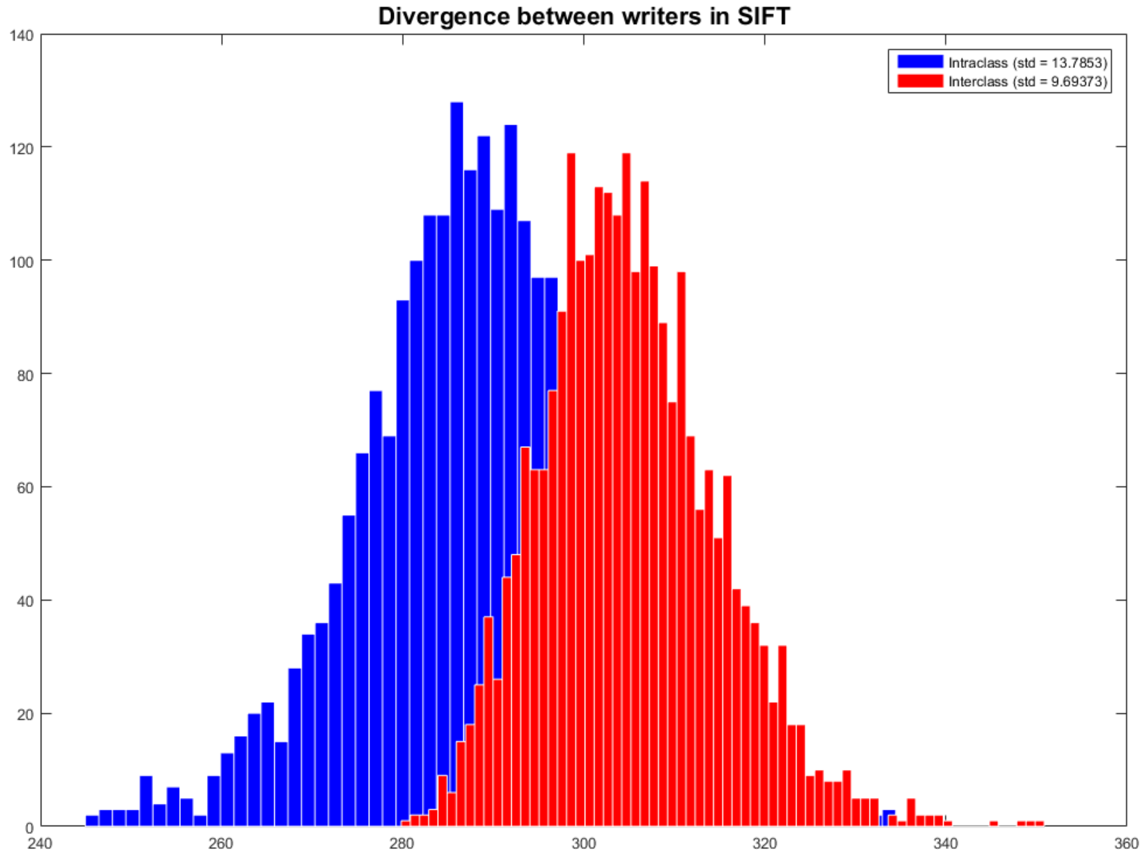


Figure 6.5: Divergence between the writers of the AHTID/MW dataset in SIFT

Nevertheless, the collection of features from a full big dataset (excluding a single writer) can be excessive. In fact, clustering such large numbers of features may lead to problems of high dimensionality and over-fitting. In order to overcome this and to ensure consistency between the DGMM and the SGMMs of the writer, a random number of feature vectors extracted from other writer's documents equal to the number of features of that writer (i.e., writer j), are selected. This random selection still covers the rest of the dataset but without causing over-fitting.

As illustrated by Fig.6.6, when a query image is presented to the system, it is first segmented into words as explained in Section 6.2.1. SIFT and RootSIFT descriptors are extracted from the segmented words of the query document. The SIFT and RootSIFT descriptors are then presented to the constructed SGMMs and DGMM of every writer. Next, instead of the conventional likelihood, a new histogram-based method is used to combine individual scores of descriptors of the same type (SIFT or RootSIFT) when a constructed GMM is applied (i.e., SGMM or DGMM). This gives an intermediate prediction score for each type of GMM. Finally, a score fusion function is used to determine the final prediction score for that writer.

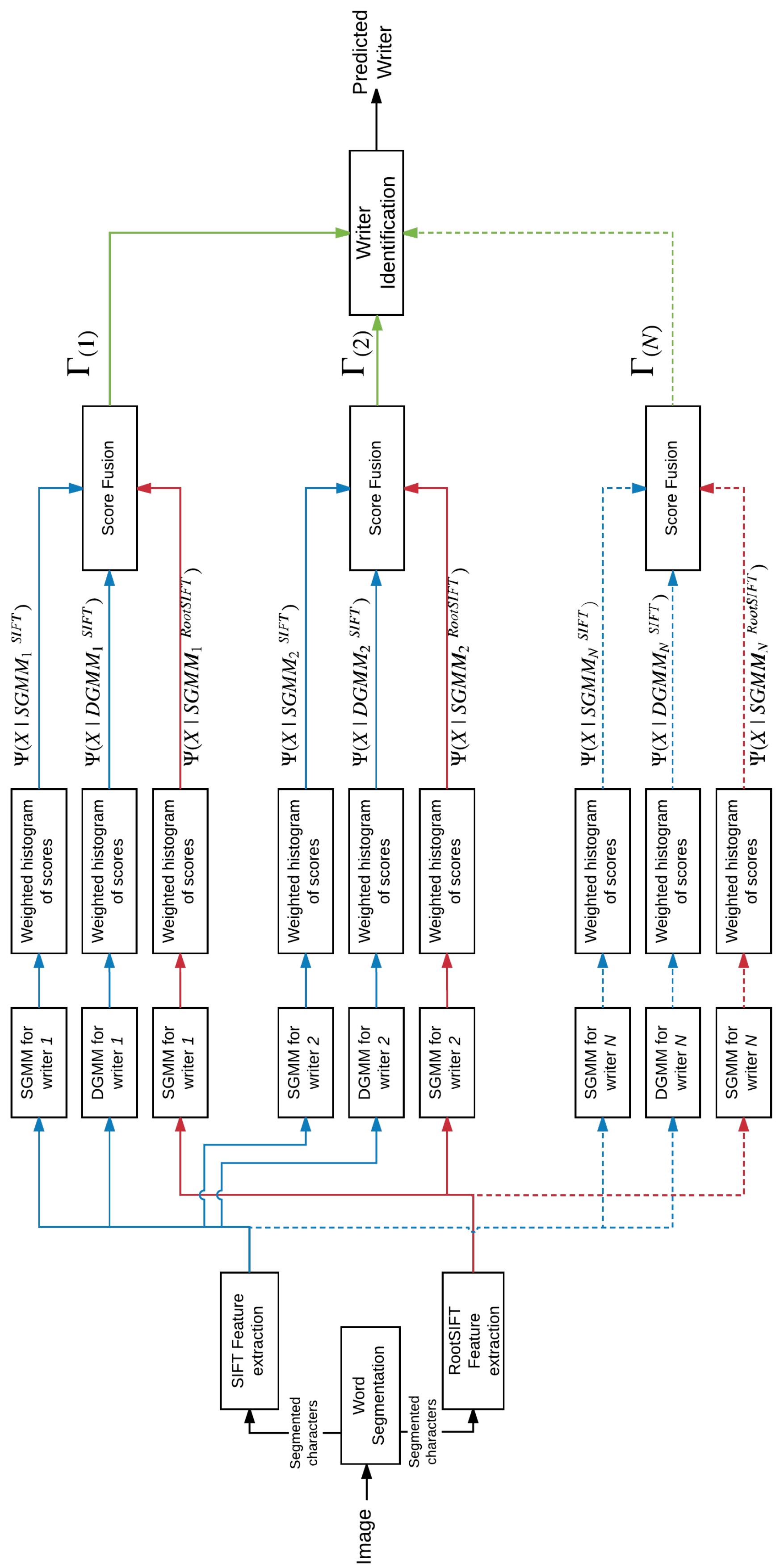


Figure 6.6: Identification of an unknown query document using the proposed system

The lowest score among all writers will correspond to the predicted writer.

6.2.4 Intermediate Prediction Score with the Weighted Histogram

The use of the negative log-likelihood as an intermediate prediction score for a query handwritten text document, as given by (6.8), can be viewed as a summation of individual negative log-probabilities, i.e., $-\log p(x_k|\lambda_j)$ where each represents the contribution of a descriptor to the negative log-likelihood. Note that, in this case, the contribution of each descriptor is treated equally. However, due to the nature of handwritten text data, different writing styles could share a few common patterns. Therefore, the dissimilar patterns should be penalized in their contribution to the intermediate score to enhance discriminability. Here, we propose a weighted histogram-based method to derive the intermediate score from individual contributions of the descriptors. In particular, the weighted factor of the histogram penalizes the large negative log-probabilities that represent dissimilarity while enforcing the the similar patterns.

Initially, the individual negative log-probabilities are analysed on the training data to determine a fixed minimum value v_{min} and maximum value v_{max} based on which histograms can be constructed. All histograms will have a constant number of bins L where the first bin covers the minimum value and the last bin covers the maximum one. This is to ensure consistency for all test images throughout the dataset. Indeed, by using these fixed minimum and maximum values and a constant number of bins, all histograms will be formed on a fixed scale. Denote by h_X^j the histogram corresponding to a query handwritten text image whose descriptors are $X = \{x_1, x_2, \cdots, x_K\}$ and presented to a GMM$_j$ for writer j (this could be a SGMM or a DGMM). It follows

$$h_X^{GMM_j}(p) = \frac{1}{K}\sum_{k=1}^{K} \delta_k^{GMM_j}(p) \; ; \; p = \{1, 2, \cdots, L\}, \tag{6.9}$$

where

$$\delta_k^j(p) = \begin{cases} 1 & v_{min} + (p-1)\,\omega \leq -\log p(x_k|\lambda_{GMM_j}) < v_{min} + p\,\omega \\ 0 & otherwise \end{cases}$$

and $\omega = \frac{v_{max} - v_{min}}{L}$ is the bin width in the histogram. λ_{GMM_j} represents the set of parameters for GMM$_j$. Theoretically, the negative log-probabilities that correspond to patterns that are similar to the ones used to generate the GMM should be very small and, therefore, the distribution of their values should fall in the first few bins of the histogram. On the other hand, the descriptors extracted from dissimilar patterns correspond to negative log-probabilities that fall in the last bins of the histogram. In the proposed method, all the bins are multiplied by an incremental value. This way, the initial bins are multiplied by a small number whereas the last bins are penalized through multiplication by a significant number in order to get clearly distinguishable. The intermediate prediction score Ψ for a query image whose descriptors are $X = \{x_1, x_2, \cdots, x_K\}$ given GMM$_j$ for writer j is provided as

$$\Psi(X|GMM_j) = \sum_{p=1}^{L} p \times h_X^{GMM_j}(p)\,. \tag{6.10}$$

Obviously, if the SGMM type is considered and the query text document, X, is indeed written by writer j, the significant histogram values will be concentrated in the first few bins. As a result, the intermediate prediction score Ψ will take a reasonably small value. On the other hand, if the query text document is not written by writer j, most of the significant histogram values will be distributed over the last bins. This will result in a large intermediate prediction score Ψ due to the imposed high weighting.

6.2.5 Score Fusion

Given a query handwritten text image, the intermediate prediction scores are calculated as explained earlier for each writer $j \in \{1, 2, \cdots, N\}$ by considering its corresponding GMMs, i.e., SGMM^{RootSIFT}, SGMM^{SIFT}, and DGMM^{SIFT}. Thus, a query image will have three scores against every writer. These intermediate scores are then fused to obtain the final prediction score Γ for each writer j as

$$\begin{aligned}\Gamma(j) \;=\;& \Psi(X|SGMM_j^{RootSIFT}) + \alpha\, \Psi(X|SGMM_j^{SIFT}) \\ & -\beta\, \Psi(X|DGMM_j^{SIFT})\,,\end{aligned} \tag{6.11}$$

where α and β are positive real numbers that act as scaling factors. This scaling is required because of the fusion of two different types of feature and GMM scores (i.e., RootSIFT and SIFT; DGMM and SGMM). The scaling parameters are determined from the training samples of each dataset by splitting the training set in two subsets, i.e. estimation and validation subsets. Using this arrangement of the training set, SGMM^{RootSIFT}, SGMM^{SIFT} and DGMM^{SIFT} for every writer are determined from the estimation subset. These models are then used to identify the 'known' samples from the validation subset. Using the known labels of the validation data, the selected values for α and β should correspond to the highest identification rate.

Once the final prediction score is calculated for each writer, the candidate writer j^* is predicted by the system as follows

$$j^* = \arg\min_{j} \Gamma(j) \tag{6.12}$$

6.3 Experimental Results and Analysis

6.3.1 Sensitivity to Model Parameters

The Top 1 accuracy achieved using SGMM$^{\text{RootSIFT}}$, SGMM$^{\text{SIFT}}$ and DGMM$^{\text{SIFT}}$ for varying number of Gaussians for all the datasets was recorded. These results are depicted in Tables 6.1 - 6.6, respectively. As can be seen, SGMM$^{\text{RootSIFT}}$ improves in terms of performance as the number of Gaussians increase, whereas the performance of SGMM$^{\text{SIFT}}$ deteriorates. Therefore, it is sensible to use a different number of Gaussians for each descriptor type at the score fusion level (Section 6.2.5).

Theoretically speaking, data can be efficiently modelled by a GMM if the number of features are significantly higher than the number of GMM parameters. However, from an implementation point of view, this was not possible for some of the datasets used in this work. Indeed, the IAM dataset, for example, has a huge variation in the number of samples per writer, and for many writers the GMM could not be built with 512 Gaussians. As a result, performance with 512 Gaussians is not shown in Tables 6.1 and 6.6, respectively.

For the IAM dataset, Table 6.7 represents the Top 1 accuracy achieved (in percentage) as a function of varying number of Gaussians used at the score fusion stage using either SIFT or RootSIFT features. The results displayed in Table 6.7 validates our observation in the sense that SIFT features are modelled more effectively with a lower number of Gaussians whereas RootSIFT features are described more effectively with a higher number of Gaussians. Thus, a judicious selection of the number of Gaussians for RootSIFT and SIFT prior to the score fusion stage can lead to high performance. As can be seen from Table 6.7 the IAM dataset performs best using 256 Gaussians for RootSIFT combined with 32 Gaussians for SIFT.

It is worth noting that the best performing number of Gaussians for SIFT and RootSIFT is in perfect agreement with that of the validation subsets used to estimate the scaling parameters (See Section 6.2.5). Therefore, from a practical perspective, the optimal number of Gaussians can be determined at the training stage on a validation subset. In the rest of the paper, the number of Gaussians used for SIFT and RootSIFT corresponds to the best performing combination

Writer model used	**Gaussians used**					
	16	**32**	**64**	**128**	**256**	**512**
SGMM$^{\text{RootSIFT}}$	68.46	83.08	89.85	91.23	89.23	-
SGMM$^{\text{SIFT}}$	30.46	25.08	21.08	17.54	13.11	-
(SGMM$^{\text{SIFT}}$ – DGMM$^{\text{SIFT}}$)	50.46	70.62	64.00	17.69	4.77	-

Table 6.1: Comparison of Top 1 accuracy achieved (in percentage) on the IAM dataset using all models at varying Gaussians.

Writer model used	**Gaussians used**					
	16	**32**	**64**	**128**	**256**	**512**
SGMM$^{\text{RootSIFT}}$	25.19	50.12	69.63	84.44	87.65	85.68
SGMM$^{\text{SIFT}}$	10.12	9.88	7.41	6.91	5.43	3.21
(SGMM$^{\text{SIFT}}$ – DGMM$^{\text{SIFT}}$)	9.14	20.49	29.63	41.01	21.48	4.13

Table 6.2: Comparison of Top 1 accuracy achieved (in percentage) on the IFN/ENIT dataset using all models at varying Gaussians.

in Tables 6.7 - 6.12, respectively.

6.3.2 Evaluation of the Score Fusion Method

In this set of experiments, the capability of the proposed fusion method, given by (6.11), of efficiently exploiting SIFT, RootSIFT, DGMM, and SGMM is demonstrated. To this end, we have considered the performance of separate descriptors SGMM$^{\text{RootSIFT}}$, SGMM$^{\text{SIFT}}$ as well as the combination of GMMs (SGMM$^{\text{SIFT}}$ - DGMM$^{\text{SIFT}}$). Results are illustrated by Table 6.13. As can be seen, the accuracy achieved with SGMM$^{\text{SIFT}}$ for each dataset has been significantly improved when DGMM$^{\text{SIFT}}$ is taken into account. Furthermore, the combination of the intermediate scores using a simple yet efficient linear function, as described by (6.11), offers a significantly higher performance. This shows that the SIFT and RootSIFT descriptors can be complementary tools for handwritten text identification.

Writer model used	Gaussians used					
	16	**32**	**64**	**128**	**256**	**512**
SGMM$^{\text{RootSIFT}}$	42.77	56.61	72.96	80.19	88.36	92.14
SGMM$^{\text{SIFT}}$	44.03	57.23	63.21	62.26	51.57	35.53
(SGMM$^{\text{SIFT}}$– DGMM$^{\text{SIFT}}$)	24.53	54.09	64.15	78.62	82.70	79.56

Table 6.3: Comparison of Top 1 accuracy achieved (in percentage) on the AHTID/MW dataset using all models at varying Gaussians.

Writer model used	Gaussians used					
	16	**32**	**64**	**128**	**256**	**512**
SGMM$^{\text{RootSIFT}}$	77.67	89.02	93.85	97.41	97.73	98.71
SGMM$^{\text{SIFT}}$	87.38	88.35	85.11	72.17	58.90	44.34
(SGMM$^{\text{SIFT}}$– DGMM$^{\text{SIFT}}$)	78.64	92.56	96.12	97.41	95.47	88.35

Table 6.4: Comparison of Top 1 accuracy achieved (in percentage) on the CVL dataset using all models at varying Gaussians.

Writer model used	Gaussians used					
	16	**32**	**64**	**128**	**256**	**512**
SGMM$^{\text{RootSIFT}}$	71.26	78.95	93.93	91.90	93.52	95.55
SGMM$^{\text{SIFT}}$	65.99	64.37	49.80	34.41	17.00	11.74
(SGMM$^{\text{SIFT}}$– DGMM$^{\text{SIFT}}$)	64.37	70.45	86.23	81.38	60.73	14.37

Table 6.5: Comparison of Top 1 accuracy achieved (in percentage) on the Firemaker dataset using all models at varying Gaussians.

Writer model used	**Gaussians used**					
	16	**32**	**64**	**128**	**256**	**512**
SGMM$^{\text{RootSIFT}}$	92.31	94.23	98.08	98.08	98.08	-
SGMM$^{\text{SIFT}}$	78.85	84.62	75.00	69.23	59.62	-
(SGMM$^{\text{SIFT}}$ – DGMM$^{\text{SIFT}}$)	82.69	88.46	96.15	94.23	84.62	-

Table 6.6: Comparison of Top 1 accuracy achieved (in percentage) on the ICDAR2011 dataset using all models at varying Gaussians.

SIFT \ Root-SIFT	**16**	**32**	**64**	**128**	**256**	**512**
16	78.77	87.38	91.23	93.85	96.15	-
32	82.77	89.69	91.54	93.54	**97.85**	-
64	84.00	90.31	91.85	92.31	90.15	-
128	79.54	86.31	88.77	88.31	86.92	-
256	11.69	18.31	25.54	32.92	45.85	-
512	-	-	-	-	-	-

Table 6.7: Top 1 accuracy achieved (in percentage) on the IAM dataset with varying number of Gaussians for the SIFT and RootSIFT features.

SIFT \ Root-SIFT	**16**	**32**	**64**	**128**	**256**	**512**
16	33.58	52.84	70.86	84.69	89.14	89.88
32	42.47	62.22	78.27	87.65	91.11	90.86
64	59.51	71.36	82.72	89.38	92.59	89.38
128	64.44	76.79	86.17	89.63	**97.28**	87.90
256	62.47	69.88	82.22	83.46	90.12	81.23
512	8.40	13.58	34.81	41.98	52.59	42.72

Table 6.8: Top 1 accuracy achieved (in percentage) on the IFN/ENIT dataset with varying number of Gaussians for the SIFT and RootSIFT features.

RootSIFT / SIFT	**16**	**32**	**64**	**128**	**256**	**512**
16	54.40	61.01	71.38	78.30	83.33	89.94
32	62.89	68.55	77.99	85.53	89.31	92.77
64	66.98	88.68	90.88	94.03	94.34	94.65
128	75.47	77.99	83.02	87.74	93.08	94.03
256	80.82	83.65	88.36	92.45	94.03	**95.60**
512	87.74	88.68	90.88	94.03	94.34	95.28

Table 6.9: Top 1 accuracy achieved (in percentage) on the AHTID/MW dataset with varying number of Gaussians for the SIFT and RootSIFT features.

RootSIFT / SIFT	**16**	**32**	**64**	**128**	**256**	**512**
16	94.17	95.79	97.73	98.06	98.71	98.71
32	94.82	95.15	96.79	98.06	98.38	98.71
64	96.12	97.41	98.06	98.38	98.71	98.71
128	97.73	98.06	98.06	98.71	98.71	**99.03**
256	98.06	98.06	98.38	98.38	98.71	98.38
512	96.76	97.09	97.09	97.41	97.73	97.73

Table 6.10: Top 1 accuracy achieved (in percentage) on the CVL dataset with varying number of Gaussians for the SIFT and RootSIFT features.

RootSIFT / SIFT	**16**	**32**	**64**	**128**	**256**	**512**
16	81.78	84.62	89.47	93.52	93.93	94.74
32	83.40	86.64	90.69	93.12	93.52	97.17
64	89.07	91.90	93.93	94.33	95.55	**97.98**
128	91.50	92.31	93.12	95.95	95.55	97.57
256	82.59	83.40	85.83	87.45	88.66	91.90
512	3.64	6.88	8.91	12.55	16.60	25.51

Table 6.11: Top 1 accuracy achieved (in percentage) on the Firemaker dataset with varying number of Gaussians for the SIFT and RootSIFT features.

SIFT \ RootSIFT	**16**	**32**	**64**	**128**	**256**	**512**
16	98.08	96.15	98.08	98.08	98.08	-
32	96.15	94.23	98.08	96.15	98.08	-
64	96.15	96.15	98.08	98.08	**100.0**	-
128	96.15	98.08	96.15	96.15	96.15	-
256	96.15	98.08	94.23	98.08	96.15	-
512	-	-	-	-	-	-

Table 6.12: Top 1 accuracy achieved (in percentage) on the ICDAR2011 dataset with varying number of Gaussians for the SIFT and RootSIFT features.

Dataset Used	**Approach used**			
	$\mathbf{SGMM}^{\text{RootSIFT}}$	$\mathbf{SGMM}^{\text{SIFT}}$	$(\mathbf{SGMM}^{\text{SIFT}}\_\mathbf{DGMM}^{\text{SIFT}})$	**Score Fusion (6.11)**
IAM	91.23	30.46	50.46	97.85
IFN/ENIT	87.65	10.12	41.01	97.28
AHTID/MW	92.14	63.21	82.70	95.60
CVL	98.71	88.35	97.41	99.03
Firemaker	95.55	65.99	86.23	97.98
ICDAR2011	98.08	84.62	96.15	100.0

Table 6.13: Comparison of Top 1 accuracy achieved (in percentage) using SGMM/DGMM models and the proposed score fusion approach.

6.3.3 Evaluation of the Weighted Histogram

As discussed in subsection 6.2.4, a query handwritten text document is represented by a number of key point descriptors where each descriptor produces its own SGMM/DGMM score. The use of the negative log-likelihood as an intermediate prediction score in the conventional approach, as given by (6.8), can be thought of as a summation of individual negative log-probabilities, where each represents the individual contribution of a descriptor to the negative log-likelihood. That is, the contribution of each descriptor is treated equally in the overall summation. Our proposed weighted histogram technique, however, is based on the fact that handwritings from the same writer should exhibit more similar textual patterns than dissimilar ones and, thus, by representing these scores in a histogram and furthermore, by penalizing the bad scores via a cost function (see (6.10)), a prominent contrast between the dissimilar handwritings can be achieved. A comparison between the proposed weighted histogram technique and the conventional negative log-likelihood one is made in Table 6.14. As can be seen, the proposed technique brings significant improvements on all datasets.

Dataset Used	**Approach used**	
	Proposed weighted histogram technique (see (6.10))	**Negative Log-likelihood (see (6.8))**
IAM	97.85	86.00
IFN/ENIT	97.28	87.41
AHTID/MW	95.60	88.05
CVL	99.03	98.38
Firemaker	97.98	91.90
ICDAR2011	100.0	98.08

Table 6.14: Top 1 accuracy achieved (in percentage) on all datasets using proposed weighted histogram-based approach versus the averaging of scores approach.

6.3.4 Comparison with Existing Works

An experimental study of the proposed system was carried out using all of the datasets described previously and the results obtained were compared with the state of the art techniques already published in the field of writer identification for their respective datasets. Comparison of the proposed system with the state of the art systems using the IAM, IFN/ENIT, AHTID/MW, CVL, Firemaker and ICDAR2011 datasets are shown in Tables 6.15, 6.16, 6.17, 6.18, 6.19 and 6.20, respectively.

Using the proposed dissimilarity based approached a Top 1 accuracy of 97.85% has been achieved on the IAM dataset, which although comparable to state of the art systems was only slightly outperformed by the system presented by Wu et al., (Wu *et al.*, 2014). Using the IFN/ENIT Arabic dataset a Top 1 accuracy of 97.28% was achieved by the proposed system which outperforms the nearest best performing system of Hannad et al., Hannad *et al.* (2016) by about 2.4%. For the AHTID/MW Arabic dataset a Top 1 accuracy of 95.60% was achieved, this result outperforms the nearest best performing system of Khan et al., (Khan *et al.*, 2016) by a margin of 8.10%. For the CVL dataset, a 99.03% of Top 1 accuracy was achieved, which was marginally outperformed by the system of Khan et al., (Khan *et al.*, 2017) by a margin of 0.57%. For the Firemaker dataset the proposed system achieved a state of the art Top 1 accuracy of 97.89%, outperforming the nearest best performing system of Wu et al., (Wu *et al.*, 2014) by 5.49%. Finally, for the cropped version of the ICDAR2011 multiple language dataset we were able to achieved a Top 1 accuracy of 100%. This dataset, although having comparatively smaller number of writers is challenging because of the multiple languages used and because of the cropped versions significantly reducing the data available per writer. However, if the number of writers were increased and made comparable to the other larger datasets we believe we may not achieve a 100% accuracy rate, but are confident that our proposed system would still fare better than the previously published systems.

6.3.5 Discussion

A significant effort was made to make the comparisons made in Section 6.3.4 transparent by clearly stating the structure and arrangement of the datasets used and by arranging the datasets in the same manner as was previously done in literature. Although some of the systems against which comparisons were made have clearly stated their dataset arrangements, many authors do not share this information. This information is valuable as changes to the dataset structure and arrangement have an impact on the performance of the system. Keeping this in view we provide the following arguments regarding our system, along with the one to one comparison.

In almost all of our comparisons, the datasets with a large number of writers have been considered. This is necessary as in real world writer identification scenarios the number of writers is a determined factor for the evaluation of the performance of any system. This was also observed by Hannad et al., that for a system of writer identification, a natural and gradual decrease in performance accuracy occurs as the number of writers are increased Hannad *et al.* (2016). Our proposed system has performed at a more than acceptable level on all datasets

System	Number of writers	Top 1 Accuracy
Bulacu and Schomaker, Bulacu & Schomaker (2007)	650	89.00%
Siddiqi and Vincent, (Siddiqi & Vincent, 2009)	650	89.00%
Siddiqi and Vincent, Siddiqi & Vincent (2010)	650	91.00%
Kumar et al., Kumar *et al.* (2014)	650	88.40%
Ghiasi and Safabakhsh, Ghiasi & Safabakhsh (2013)	650	93.70%
Bertolini et al. , Bertolini *et al.* (2013)	650	96.70%
Khalifa et al., Khalifa *et al.* (2015)	650	92.00%
Jain and Doermann, (Jain & Doermann, 2014)	657	94.70%
Hannad et al., Hannad *et al.* (2016)	657	89.50%
Brink et al., (Brink *et al.*, 2012)	657	97.00%
Schomaker and Bulacu, Schomaker & Bulacu (2004)	657	82.50 %
Khan et al., (Khan *et al.*, 2017)	650	97.20%
He et al., (He *et al.*, 2015)	650	91.10%
Wu et al., (Wu *et al.*, 2014)	**650**	**98.50%**
Proposed system	650	97.85%

Table 6.15: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the IAM dataset.

System	Number of writers	Top 1 Accuracy
Bulacu and Schomaker, (Bulacu *et al.*, 2007)	350	88.00%
Chawki et al., (Chawki & Labiba, 2010)	130	82.00%
Djeddi et al., (Djeddi & Souici-Meslati, 2011)	130	84.23%
Abdi and Khemakhem, (Abdi & Khemakhem, 2012)	100	85.00%
Abdi and Khemakhem, Abdi & Khemakhem (2015)	411	90.02%
Hannad et al., Hannad *et al.* (2016)	411	94.89%
Khan et al., (Khan *et al.*, 2017)	411	76.00%
Proposed system	**411**	**97.28%**

Table 6.16: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the IFN/ENIT dataset.

System	Number of writers	Top 1 Accuracy
Slimane and Margner, (Slimane & Margner, 2014)	53	69.40%
Schomaker and Bulacu, Schomaker & Bulacu (2004)	53	66.40%
Hannad et al., Hannad *et al.* (2016)	53	77.30%
Khan et al., (Khan *et al.*, 2016)	53	87.50%
Khan et al., (Khan *et al.*, 2017)	53	71.60%
Proposed system	**53**	**95.60%**

Table 6.17: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the AHTID/MW dataset.

System	Number of writers	Top 1 Accuracy
Fiel and Sablatnig, (Fiel & Sablatnig, 2013)	309	97.80%
Jain and Doermann, (Jain & Doermann, 2014)	310	99.40%
Christlein et al., (Christlein *et al.*, 2017)	310	99.20%
Fiel and Sablatnig, (Fiel & Sablatnig, 2015)	309	98.90%
Schomaker and Bulacu, Schomaker & Bulacu (2004)	310	81.80%
Hannad et al., Hannad *et al.* (2016)	310	96.20%
Khan et al., (Khan *et al.*, 2017)	**310**	**99.60%**
Proposed system	310	99.03%

Table 6.18: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the CVL dataset.

System	Number of writers	Top 1 Accuracy
Bulacu and Schomaker, Bulacu & Schomaker (2007)	250	83.00%
Li and Ding, (Li & Ding, 2009)	250	78.00%
Brink et al., (Brink *et al.*, 2012)	250	86.00%
Ghiasi et al., (Ghiasi & Safabakhsh, 2013)	250	91.80%
He et al., (He *et al.*, 2015)	250	89.80%
Wu et al., (Wu *et al.*, 2014)	250	92.40%
Khan et al., (Khan *et al.*, 2017)	250	89.47%
Proposed system	**250**	**97.98%**

Table 6.19: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the Firemaker dataset.

System	Number of writers	Top 1 Accuracy
ECNU method, (Louloudis *et al.*, 2011)	26 (cropped)	65.90%
QUQA-a method, (Louloudis *et al.*, 2011)	26 (cropped)	74.00%
QUQA-b method, (Louloudis *et al.*, 2011)	26 (cropped)	67.30%
TSINGHUA method, (Louloudis *et al.*, 2011)	26 (cropped)	90.90%
GWU method, (Louloudis *et al.*, 2011)	26 (cropped)	74.00%
CS-UMD method, (Louloudis *et al.*, 2011)	26 (cropped)	66.80%
TEBESSA method, (Louloudis *et al.*, 2011)	26 (cropped)	87.50%
MCS-NUST method, (Louloudis *et al.*, 2011)	26 (cropped)	82.20%
Wu et al., (Wu *et al.*, 2014)	26 (cropped)	95.20%
Khan et al., (Khan *et al.*, 2017)	26 (cropped)	82.69%
Proposed system	**26 (cropped)**	**100.0%**

Table 6.20: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the ICDAR2011 dataset.

having large number of writers. However, the major limitation of our system is that since it relies on GMM clustering to generate a writer model, it requires a fairly large number of features to generate a strong predictor model of any writer. This explains the incorrect predictions achieved when using the IAM dataset as the writers having limited training data were unable to provide a strong and correct prediction.

Furthermore, in the case of the Arabic datasets, it was observed by Balacu et al., (Bulacu *et al.*, 2007) and Khan et al. (Khan *et al.*, 2017) that systems that tend to perform better on scripts such as Roman and Latin fail to perform acceptably when applied to an Arabic dataset. It was concluded that due to the Arabic writing style, identification of Arabic handwriting is a more challenging task than identification of writers in other scripts. Keeping in view this observation, our proposed system was able to perform well irrespective of the script used.

The main contributions of this work can be summarised as: (i) Dissimilarity Gaussian Mixture Models are introduced. Along with two similarity GMMs created for every writer, a DGMM is also constructed for describing the contrast between different writers in the dataset. The combination of DGMM with SGMM has been shown to bring significant improvements over the overall identification rate. (ii) Because a handwritten text is described by a number of key point descriptors where each descriptor has its own SGMM/DGMM contribution, a new weighted histogram method is proposed to derive the intermediate prediction score from the set of individual key point contributions. The idea of the weighted histogram relies on the fact that handwritings of the same writer should exhibit more similar textual patterns than dissimilar ones. Therefore, by penalizing the bad contributions with a cost function, the identification rate can be significantly enhanced. (iii) SIFT and RootSIFT descriptors are effectively exploited in a joined SGMM/DGMM system where a simple but efficient score fusion method has been proposed accordingly.

6.4 Conclusion

In this paper, an offline handwritten text identification system has been proposed. The concept of similarity and dissimilarity GMMs has been introduced and incorporated in the proposed system. Furthermore, SIFT and RootSIFT descriptors have been extracted from handwritten text images. These descriptors are then used to generate similarity and dissimilarity GMMs for each writer. Interestingly, the use of both SIFT and RootSIFT descriptors combined together in a single system has proven to be efficient on handwritten text data which suggests that the two features are complementary rather than redundant for handwritten text identification. Given a query text image, a GMM produces an intermediate prediction score via a new weighted histogram-based method for each writer. This has been shown to perform significantly better than the conventional averaging of the negative log-likelihood scores, because the contribution of irrelevant descriptors is penalized by the weighting process. Intermediate prediction scores are then efficiently fused using a linear function to obtain the final prediction score. Assessed on a number of handwritten text datasets through intensive experiments, the proposed system has been shown to operate remarkably well with different handwritten languages. Experiments have also shown the superiority of the proposed system over state-of-the-art techniques.

The next chapter concludes this thesis with the conclusions drawn from this research and also makes suggestions for possible future work.

Chapter 7

Conclusion

The aim of this chapter is to review and summarize the main contributions made in this thesis in relation to offline text independent hand writer identification. This study has reviewed the advancements carried out in the field of writer identification and proposed solutions to further contribute to the field.

In general, automatic identification of writers from offline documents is a challenging task due to the inherent difficulties experienced in pattern recognition. As explained in Chapter 1, the hand writer identification process consists of four stages: acquisition of images, pre-processing acquired images, extraction of features and finally classification. The main contributions of this study relates to the pre-processing, feature extraction and classification stages. This study contributes to the field of writer identification by performing an in depth literature review of previously published works and by proposing and implementing new segmentation and identification algorithms, that have furthered the knowledge of this field. By performing this research, the aims and objectives outlined in Chapter 1 have been achieved.

7.1 Summary of Contributions

- In Chapter 3, the problem of segmenting overlapping sub-words within hand written Arabic words was addressed. It was observed that the conventional method of projection profiles for segmentation of English and Latin words could not be applied effectively to the Arabic language. Projection profile method relies on the gap between the letters to effectively segment them, this cannot be applied in Arabic script as one letter is usually written before the previous one ends. Thus, to the projection profile method two or more characters may look like a single long word. To address these issues, a new method of segmenting sub-words within Arabic words was proposed. Segmentation was achieved directly as a result of the pre-processing step of binarization. The words were binarized at minutely increasing threshold levels to achieve clusters of connected components. These clusters, when plotted on an area graph allowed for diacritics to be separated and the sub-words to be effectively extracted. The plotted data was assumed to be normally distributed and therefore, the clusters could be automatically extracted using the 95% confidence interval. The proposed method is unaffected by the skew of the document and text lines and was also shown to be robust in the presence of noise. The proposed method was applied on 537 randomly selected words from the AHTID/MW database, of which 95.3% of the words were correctly segmented.

- In Chapter 4, a segmentation free hand writer identification system was proposed using textural features. LTP features were extracted at various scales in order to best capture the most dominant features from the hand written text. Thus a multi scale Local Ternary Pattern Histogram feature was proposed that extended the traditional LTP to a multi scale representation. The representation of images in a multi scale level came at the cost of high dimensionality. In order to avoid the problems associated with high dimensionality, SR-KDA was applied for the purpose of dimensionality reduction. An identifier model was created for every LTP scale and identification was performed by aggregating the predicted results of all these models. The proposed system was evaluated on the Arabic AHTID/MW and IAM datasets. On both datasets, the proposed system demonstrated promising results compared to previously published works.

- Chapter 5 presents a novel Bagged Discrete Cosine Transform (BDCT) approach for offline text independent writer identification. DCT features were extracted from small overlapping blocks. DCT was used because of its reputation for being a good and robust feature extractor for the purpose of image matching. To overcome memory limitations caused by an excessive amount of DCT features, unique random DCT features were selected from every image. These randomly selected features were then clustered using k-means clustering to generate a universal codebook which could then be used for generating descriptors for every image. SR-KDA was utilized to reduce the dimensionality of the feature space as well as decreasing the intra-class variance. SR-KDA produced a predictor model for that random selection of features that could be used to identify an unknown sample of text. Bootstrap aggregation was then used for the ensemble of these predictor models. The proposed system was applied on four challenging datasets, covering English and Arabic scripts and on all datasets promising results were shown when compared with previously published works (except for the binary IFN/ENIT dataset).

Furthermore, the IAM and AHTID datasets were subjected to two types of distortions to test the robustness of the proposed system. When compared with previously published works, the proposed system greatly outperformed all other systems when applied on noisy and distorted images.

- Chapter 6 presents an efficient handwriting identification system that combines SIFT and RootSIFT descriptors in a set of Gaussian mixture models (GMM). In particular, a new concept of similarity and dissimilarity Gaussian mixture models (SGMM and DGMM) was introduced. While a SGMM is constructed for every writer to describe the intra-class similarity that is exhibited between the handwritten texts of the same writer, the DGMM represents the contrast that exists between the writer's style on one hand and other different handwriting styles on the other hand. Furthermore, because the handwritten text is described by a number of key point descriptors where each descriptor leads to a SGMM/DGMM score, a new weighted histogram method was proposed to derive the intermediate prediction score for each writer's GMM. The idea of weighted histogram relies on the fact that handwritings of the same writer should exhibit more similar textual patterns than dissimilar ones and, hence, by penalizing the bad scores with a cost function, the identification rate can be significantly enhanced. The proposed system was evaluated on six publicly available datasets of multiple languages, three English, two Arabic and one Hybrid language dataset, and was also validated against state-of-the-art systems.

7.2 Future Work

- The writer identification system proposed in Chapter 4 was evaluated on a 100 writer identification task of the IAM dataset, a benchmark subset that was arranged for the purpose of writer identification by the authors of the dataset. However, during the course of this work the 100 writer identification task was withdrawn by the authors. At the current level of research, the performance of a writer identification system can only be assessed after it has been evaluated on a large number of writers, therefore along with the 100 writer result, efforts were made to apply the system on the full IAM dataset of 650 writers. It was observed that the proposed system, in its current form, does not scale well to the full dataset, therefore we propose to introduce some level of segmentation before feature extraction as was done by (Hannad *et al.*, 2016) to improve the overall accuracy when using LTP.

- The system proposed in chapter 5 utilizes DCT as its main feature extractor, due to which the system cannot perform well on documents presented in binary form. This is a weakness that can be addressed in future work by combining the DCT features with other local features that capture shape rather than the frequency content. Also, because SR-KDA uses all training samples to optimize the parameters of the feature mapping function, adding a new entry (writer) to the database in practice would require a new estimation of the parameters and this may be computationally expensive, especially, when the number of existing writers in the database is significantly large.

- The fusion of scores in Chapter 6 is made possible with the help of scaling factors which are determined by splitting the training set in two subsets, i.e. estimation and validation subsets. For future work we intend to automate the process of determining the scaling factors,α and β. This would involve training a positive and negative population from the training samples of all writers of a dataset. These populations would then be fed to a binary SVM classifier to obtain the alpha vector and the support vectors, the coefficients of the support vectors would provide us with the α and β values for that dataset.

- Deep learning is currently the fastest growing field in machine learning and might contribute significantly in the field of handwritten text identification. Recent research has demonstrated that deep convolutional neural network (CNN's) have significantly improved the state-of-the-art results in object detection (Ren *et al.*, 2015), image classification (He *et al.*, 2016) and face recognition (Sun *et al.*, 2015). Deep CNN's require a lot of data to train and usually pre-trained networks (like AlexNet) are re-trained using transfer learning for object detection or classification. However, this might prove to be challenging for hand writer identification since no pre-trained network for writer recognition exists and a single writer may not have enough training data to accurately model an identifier (as is the case with many writers in the IAM dataset). For such cases where input data is insufficient, the input images can be divided into overlapping patches (as was done in (Xing & Qiao, 2016) and (Yang *et al.*, 2016)) or networks pre-trained on other languages can be used since different languages share common features when identifying writers (Xing & Qiao, 2016).

References

Abdi, Mohamed Nidhal, & Khemakhem, Maher. 2012. Arabic writer identification and verification using template matching analysis of texture. *Pages 592–597 of: Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on*. IEEE.

Abdi, Mohamed Nidhal, Khemakhem, Maher, & Ben-Abdallah, H. 2010. Off-line text-independent arabic writer identification using contour-based features. *International Journal of Signal and Image Processing*, **1**(1), 4–11.

Abed, H. E., Kherallah, M., Mï¿œrgner, V., & Alimi, A. M. 2011. On-line Arabic handwriting recognition competition, ADAB database and participating systems. *International Journal on Document Analysis and Recognition*, **14**(1), 15–23.

Al-Dmour, Ayman, & Abu Zitar, Raed. 2007. Arabic writer identification based on hybrid spectral–statistical measures. *Journal of Experimental & Theoretical Artificial Intelligence*, **19**(4), 307–332.

Al-Ma'adeed, Somaya, Elliman, Dave, & Higgins, Colin A. 2002. A data base for Arabic handwritten text recognition research. IEEE.

Al-Ma'adeed, Somaya, Mohammed, Eman, Al Kassis, Dori, & Al-Muslih, Fatma. 2008. Writer identification using edge-based directional probability distribution features for arabic words. *Pages 582–590 of: Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*. IEEE.

Al Maadeed, Somaya, Ayouby, Wael, Hassaï¿œne, Abdelï¿œali, & Aljaam, Jihad Mohamad. 2012. Quwi: An arabic and english handwriting dataset for offline writer identification. IEEE.

Alamri, Huda, Sadri, Javad, Suen, Ching Y., & Nobile, Nicola. 2008. A novel comprehensive database for Arabic off-line handwriting recognition. vol. 8.

Amara, Najoua Essoukri Ben, Mazhoud, Omar, Bouzrara, Noura, & Ellouze, Noureddine. 2005. ARABASE: A Relational Database for Arabic OCR Systems. *Int. Arab J. Inf. Technol.*, **2**(4), 259–266.

Arandjelović, Relja, & Zisserman, Andrew. 2012. Three things everyone should know to improve object retrieval. *Pages 2911–2918 of: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE.

Arora, Hina, Lee, Sangjik, Srihari, Sargur N, & Cha, Sunk-Hyuk. 2002. Individuality of handwriting. *Journal of forensic science*, **47**(4), 1–17.

Ashbourn, Julian. 2014. *Biometrics in the new world: The cloud, Mobile technology and Pervasive identity*. Springer Science & Business Media.

Baghshah, M Soleymani, Shouraki, S Bagheri, & Kasaei, S. 2006. A novel fuzzy classifier using fuzzy LVQ to recognize online persian handwriting. *Pages 1878–1883 of: Information and Communication Technologies, 2006. ICTTA'06. 2nd*, vol. 1. IEEE.

Ballard, Lucas, Lopresti, Daniel, & Monrose, Fabian. 2006. Evaluating the security of handwriting biometrics. *Pages – of: Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

Baudat, Gaston, & Anouar, Fatiha. 2000. Generalized discriminant analysis using a kernel approach. *Neural computation*, **12**(10), 2385–2404.

Bensefia, Ameur, Paquet, Thierry, & Heutte, Laurent. 2005a. Handwritten document analysis for automatic writer recognition. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, **5**(2), 72–86.

Bensefia, Ameur, Paquet, Thierry, & Heutte, Laurent. 2005b. A writer identification and verification system. *Pattern Recognition Letters*, **26**(13), 2080–2092.

Bertolini, Diego, Oliveira, Luiz S, Justino, E, & Sabourin, Robert. 2013. Texture-based descriptors for writer identification and verification. *Expert Systems with Applications*, **40**(6), 2069–2080.

Breiman, Leo. 1996. Bagging predictors. *Machine learning*, **24**(2), 123–140.

Brink, AA, Smit, J, Bulacu, ML, & Schomaker, LRB. 2012. Writer identification using directional ink-trace width measurements. *Pattern Recognition*, **45**(1), 162–171.

Bulacu, Marius, & Schomaker, Lambert. 2005. A comparison of clustering methods for writer identification and verification. *Pages 1275–1279 of: Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE.

Bulacu, Marius, & Schomaker, Lambert. 2007. Text-independent writer identification and verification using textural and allographic features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(4), 701–717.

Bulacu, Marius, Schomaker, Lambert, & Brink, Axel. 2007. Text-independent writer identification and verification on offline arabic handwriting. *Pages 769–773 of: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE.

Cai, Deng, He, Xiaofei, & Han, Jiawei. 2007. Efficient kernel discriminant analysis via spectral regression. *Pages 427–432 of: Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE.

Cai, Deng, He, Xiaofei, & Han, Jiawei. 2011. Speed up kernel discriminant analysis. *The VLDB Journal*, **20**(1), 21–33.

Chan, Chi-Ho, Kittler, Josef, & Messer, Kieron. 2007. *Multi-scale local binary pattern histograms for face recognition*. Springer.

Chawki, Djeddi, & Labiba, Souici-Meslati. 2010. A texture based approach for Arabic Writer Identification and Verification. *Pages 115–120 of: Machine and Web Intelligence (ICMWI), 2010 International Conference on*. IEEE.

Chen, Jin. 2012. *Handwritten biometric systems and their robustness evaluation: a survey*. Tech. rept.

Christlein, Vincent, Bernecker, David, Honig, Florian, & Angelopoulou, Elli. 2014. Writer identification and verification using GMM supervectors. *Pages 998–1005 of: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE.

Christlein, Vincent, Bernecker, David, Hönig, Florian, Maier, Andreas, & Angelopoulou, Elli. 2017. Writer Identification Using GMM Supervectors and Exemplar-SVMs. *Pattern Recognition*, **63**, 258–267.

Davis, Tom. 2007. The practice of handwriting identification. *The Library*, **8**(3), 251–276.

Dempster, Arthur P, Laird, Nan M, & Rubin, Donald B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Dinges, L., Al-Hamadi, A., & Elzobi, M. 2013 (Aug). A Locale Group Based Line Segmentation Approach for Non Uniform Skewed and Curved Arabic Handwritings. *Pages 803–806 of: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*.

Djeddi, Chawki, & Souici-Meslati, Labiba. 2011. Artificial immune recognition system for Arabic writer identification. *Pages 159–165 of: Innovation in Information & Communication Technology (ISIICT), 2011 Fourth International Symposium on*. IEEE.

Djeddi, Chawki, Siddiqi, Imran, Souici-Meslati, Labiba, & Ennaji, Abdellatif. 2013. Text-independent writer recognition using multi-script handwritten texts. *Pattern Recognition Letters*, **34**(10), 1196–1202.

Fiel, Stefan, & Sablatnig, Robert. 2012. Writer retrieval and writer identification using local features. *Pages 145–149 of: Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE.

Fiel, Stefan, & Sablatnig, Robert. 2015. Writer Identification and Retrieval Using a Convolutional Neural Network. *Pages 26–37 of: Computer Analysis of Images and Patterns*. Springer.

Fornés, Alicia, Lladós, Josep, Sánchez, Gemma, & Bunke, Horst. 2008. Writer identification in old handwritten music scores. *Pages 347–353 of: Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*. IEEE.

Fornes, Alicia, Llados, Josep, Sanchez, Gemma, & Bunke, Horst. 2009. On the use of textural features for writer identification in old handwritten music scores. *Pages 996–1000 of: 2009 10th International Conference on Document Analysis and Recognition*. IEEE.

Fukunaga, Keinosuke. 2013. *Introduction to statistical pattern recognition*. Academic press.

Gazzah, Sami, & Amara, N Ben. 2007. Arabic handwriting texture analysis for writer identification using the dwt-lifting scheme. *Pages 1133–1137 of: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE.

Gazzah, Sami, & Essoukri Ben Amara, Najoua. 2006. Writer identification using modular MLP classifier and genetic algorithm for optimal features selection. *Advances in Neural Networks-ISNN 2006*, 271–276.

Ghiasi, Golnaz, & Safabakhsh, Reza. 2013. Offline text-independent writer identification using codebook and efficient code extraction methods. *Image and Vision Computing*, **31**(5), 379–391.

Gracia, V. 2006. State of the art in biometrics research and market survey. *HUMABIO Project (EU FP6 contract no 026990), Deliverable*, –.

Grosicki, Emmanuï¿œle, Carre, Matthieu, Brodin, Jean-Marie, & Geoffrois, Edouard. 2008. RIMES evaluation campaign for handwritten mail processing. Concordia University.

Hannad, Yaacoub, Siddiqi, Imran, & El Kettani, Mohamed El Youssfi. 2016. Writer identification using texture descriptors of handwritten fragments. *Expert Systems with Applications*, **47**, 14–22.

Hanusiak, Regiane Kowalek, Oliveira, Luiz S, Justino, E, & Sabourin, Robert. 2012. Writer verification using texture-based features. *International Journal on Document Analysis and Recognition (IJDAR)*, **15**(3), 213–226.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. 2016. Deep residual learning for image recognition. *Pages 770–778 of: Proceedings of the IEEE conference on computer vision and pattern recognition*.

He, Sheng, Wiering, Marco, & Schomaker, Lambert. 2015. Junction detection in handwritten documents and its application to writer identification. *Pattern Recognition*, **48**(12), 4036–4048.

Helli, Behzad, & Moghaddam, Mohsen Ebrahimi. 2008. A text-independent Persian writer identification system using LCS based classifier. *Pages 203–206 of: 2008 IEEE International Symposium on Signal Processing and Information Technology*. IEEE.

Hertel, Caroline, & Bunke, Horst. 2003. A set of novel features for writer identification. *Pages 679–687 of: International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer.

Hu, Yongjie, Yang, Wenming, & Chen, Youbin. 2014. Bag of features approach for offline text-independent Chinese writer identification. *Pages 2609–2613 of: Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE.

Impedovo, Donato, & Pirlo, Giuseppe. 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.

Jain, Anil, Bolle, Ruud, & Pankanti, Sharath. 2006. *Biometrics: personal identification in networked society*. Vol. 479. Springer Science & Business Media.

Jain, Anil K, Griess, Friederike D, & Connell, Scott D. 2002. On-line signature verification. *Pattern recognition*, **35**(12), 2963–2972.

Jain, Anil K, Ross, Arun, & Prabhakar, Salil. 2004. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, **14**(1), 4–20.

Jain, Rajiv, & Doermann, David. 2013. Writer identification using an alphabet of contour gradient descriptors. *Pages 550–554 of: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE.

Jain, Rajiv, & Doermann, David. 2014. Combining Local Features For Offline Writer Identification. *Pages – of: 2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE.*

Jin, Lianwen, Gao, Yan, Liu, Gang, Li, Yunyang, & Ding, Kai. 2011. SCUT-COUCH2009–a comprehensive online unconstrained Chinese handwriting database and benchmark evaluation. *International Journal on Document Analysis and Recognition (IJDAR)*, **14**(1), 53–64.

Johnson, Oliver Thomas. 2004. *Information theory and the central limit theorem*. World Scientific.

Kavallieratou, Ergina, Liolios, Nikos, Koutsogeorgos, E., Fakotakis, Nikos, & Kokkinakis, G. 2001. The GRUHD database of Greek unconstrained handwriting. IEEE.

Khalifa, E, Al-maadeed, S, Tahir, MA, Bouridane, A, & Jamshed, A. 2015. Off-line writer identification using an ensemble of grapheme codebook features. *Pattern Recognition Letters*, **59**, 18–25.

Khan, Faraz, Bouridane, Ahmed, Khelifi, Fouad, Almotaeryi, Rasheed, & Almaadeed, Sumaya. 2014. Efficient segmentation of sub-words within handwritten arabic words. *Pages 684–689 of: Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*. IEEE.

Khan, Faraz Ahmad, Tahir, Muhammad Atif, Khelifi, Fouad, & Bouridane, Ahmed. 2016. Offline text independent writer identification using ensemble of multi-scale local ternary pattern histograms. *Pages 1–6 of: Visual Information Processing (EUVIP), 2016 6th European Workshop on*. IEEE.

Khan, Faraz Ahmad, Tahir, Muhammad Atif, Khelifi, Fouad, Bouridane, Ahmed, & Almotaeryi, Resheed. 2017. Robust off-line text independent writer identification using bagged discrete cosine transform features. *Expert Systems with Applications*, **71**, 404–415.

Kharma, Nawwaf, Ahmed, Maher, & Ward, Rabab. 1999. A new comprehensive database of handwritten Arabic words, numbers, and signatures used for OCR testing. vol. 2. IEEE.

Kırlı, Önder, & Gülmezoğlu, M Bilginer. 2012. Automatic writer identification from text line images. *International Journal on Document Analysis and Recognition (IJDAR)*, **15**(2), 85–99.

Kittler, Josef, Hatef, Mohamad, Duin, Robert PW, & Matas, Jiri. 1998. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **20**(3), 226–239.

Kleber, Florian, Fiel, Stefan, Diem, Markus, & Sablatnig, Robert. 2013. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. *Pages 560–564 of: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE.

Kumar, Rajesh, Chanda, Bhabatosh, & Sharma, JD. 2014. A novel sparse model based forensic writer identification. *Pattern Recognition Letters*, **35**, 105–112.

Lawgali, A., Bouridane, A., Angelova, M., & Ghassemlooy, Z. 2011. Automatic segmentation for Arabic characters in handwriting documents. *Pages 3529–3532 of: Image Processing (ICIP), 2011 18th IEEE International Conference on*.

Leedham, Graham, & Chachra, Sumit. 2003. Writer identification using innovative binarised features of handwritten numerals. *Pages 413–416 of: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. IEEE.

Li, Bangy, & Tan, Tieniu. 2009. Online text-independent writer identification based on temporal sequence and shape codes. *Pages 931–935 of: Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE.

Li, Xin, & Ding, Xiaoqing. 2009. Writer identification of chinese handwriting using grid microstructure feature. *Pages 1230–1239 of: International Conference on Biometrics*. Springer.

Liu, Cheng-Lin, Yin, Fei, Wang, Da-Han, & Wang, Qiu-Feng. 2011. CASIA online and offline Chinese handwriting databases. IEEE.

Louloudis, Georgios, Stamatopoulos, Nikolaos, & Gatos, Basilios. 2011. ICDAR 2011 writer identification contest. *Pages 1475–1479 of: 2011 International Conference on Document Analysis and Recognition*. IEEE.

Louloudis, Georgios, Gatos, Basilis, & Stamatopoulos, Nikolaos. 2012. Icfhr 2012 competition on writer identification challenge 1: Latin/greek documents. *Pages 829–834 of: Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*. IEEE.

Lowe, David G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60**(2), 91–110.

Mahmoud, Sabri A., Ahmad, Irfan, Alshayeb, Mohammad, Al-Khatib, Wasfi G., Parvez, Mohammad Tanvir, Fink, Gernot A., Mï¿œrgner, Volker, & El Abed, Haikal. 2012. KHATT: Arabic offline handwritten text database. IEEE.

Manohar, V., Vitaladevuni, S.N., Cao, Huaigu, Prasad, R., & Natarajan, P. 2011 (Sept). Graph Clustering-Based Ensemble Method for Handwritten Text Line Segmentation. *Pages 574–578 of: Document Analysis and Recognition (ICDAR), 2011 International Conference on*.

Margner, V., & Abed, H.E. 2012. *Guide to OCR for Arabic Scripts*. Springer.

Marti, U-V, & Bunke, Horst. 2002. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, **5**(1), 39–46.

Marti, U-V, Messerli, R, & Bunke, Horst. 2001. Writer identification using text line based features. *Pages 101–105 of: Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE.

Mendenhall, William, Beaver, Robert J, & Beaver, Barbara M. 2012. *Introduction to probability and statistics*. Cengage Learning.

Mezghani, Amine, Kanoun, Slim, Khemakhem, Mahdi, & Abed, Haikal El. 2012. A database for arabic handwritten text image recognition and writer identification. *Pages 399–402 of: Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*. IEEE.

Mitrea, Mihai, PrÃªteux, FranÃ§oise, Vlad, Adriana, & Fetita, Catalin. 2004. The 2D-DCT coefficient statistical behaviour: a comparative analysis on different types of image sequences. *Journal of Optoelectronics and Advanced Materials*, **6**(1), 95–102.

Namboodiri, Anoop, & Gupta, Sachin. 2006. Text independent writer identification from online handwriting. *Pages – of: Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

Nejad, F, & Rahmati, M. 2007. A new method for writer identification and verification based on Farsi/Arabic handwritten texts. *Pages 829–833 of: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. IEEE.

Newell, Andrew J, & Griffin, Lewis D. 2014. Writer identification using oriented basic image features and the delta encoding. *Pattern Recognition*, **47**(6), 2255–2265.

Nikulin, Mikhail S. 2001. Hellinger distance. *Encyclopedia of mathematics*, **151**, –.

Nister, David, & Stewenius, Henrik. 2006. Scalable recognition with a vocabulary tree. *Pages 2161–2168 of: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE.

Ojala, Timo, PietikÃ€inen, Matti, & Harwood, David. 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, **29**(1), 51–59.

Ojala, Timo, Pietikäinen, Matti, & Mäenpää, Topi. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **24**(7), 971–987.

Osman, Y. 2013. Segmentation algorithm for Arabic handwritten text based on contour analysis. *Pages 447–452 of: Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*.

Otsu, Nobuyuki. 1975. A threshold selection method from gray-level histograms. *Automatica*, **11**(285-296), 23–27.

Otsu, Nobuyuki. 1979. A Threshold Selection Method from Gray-Level Histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, **9**(1), 62–66.

Pechwitz, Mario, Maddouri, S Snoussi, Märgner, Volker, Ellouze, Noureddine, Amiri, Hamid, *et al.* 2002. IFN/ENIT-database of handwritten Arabic words. *Pages 127–136 of: Proc. of CIFED*, vol. 2. Citeseer.

Rafiee, Ali, & Motavalli, Hamidreza. 2007. Off-line writer recognition for farsi text. *Pages 193–197 of: Artificial Intelligence-Special Session, 2007. MICAI 2007. Sixth Mexican International Conference on*. IEEE.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, & Sun, Jian. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Pages 91–99 of: Advances in neural information processing systems*.

Reynolds, Douglas A. 1995. Speaker identification and verification using Gaussian mixture speaker models. *Speech communication*, **17**(1), 91–108.

Reynolds, Douglas A, Quatieri, Thomas F, & Dunn, Robert B. 2000. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, **10**(1), 19–41.

Riera, Alessandro, Soria-Frisch, Aureli, Caparrini, Mario, Cester, Ivan, & Ruffini, Giulio. 2009. Multimodal physiological biometrics authentication. *Biometrics: Theory, Methods, and Applications*, 461–482.

Robinson, Jonathan, & Kecman, Vojislav. 2003. Combining support vector machine learning with the discrete cosine transform in image compression. *Neural Networks, IEEE Transactions on*, **14**(4), 950–958.

Safabaksh, Reza, Ghanbarian, Ali Reza, & Ghiasi, Golnaz. 2013. HaFT: A handwritten Farsi text database. IEEE.

Sarfraz, M. 2005. *Computer-Aided Intelligent Recognition Techniques and Applications*. Wiley.

Sayood, K. 2012. *Introduction to Data Compression*. Morgan Kaufmann.

Schlapbach, Andreas, & Bunke. 2006. Off-linewriter identification using Gaussian mixture models. *Pages 992–995 of: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE.

Schlapbach, Andreas, & Bunke, Horst. 2004. Using HMM based recognizers for writer identification and verification. *Pages 167–172 of: Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*. IEEE.

Schlapbach, Andreas, & Bunke, Horst. 2007. A writer identification and verification system using HMM based recognizers. *Pattern Analysis and Applications*, **10**(1), 33–43.

Schlapbach, Andreas, Kilchherr, Vivian, & Bunke, Horst. 2005. Improving writer identification by means of feature selection and extraction. *Pages 131–135 of: Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE.

Schomaker, Lambert, & Bulacu, Marius. 2004. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(6), 787–798.

Schomaker, Lambert, & Vuurpijl, Louis. 2000. Forensic writer identification: A benchmark data set and a comparison of two systems [internal report for the Netherlands Forensic Institute]. –.

Schomaker, Lambert, Bulacu, Marius, & Franke, Katrin. 2004. Automatic writer identification using fragmented connected-component contours. *Pages 185–190 of: Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*. IEEE.

Schomaker, Lambert, Franke, Katrin, & Bulacu, Marius. 2007. Using codebooks of fragmented connected-component contours in forensic and historic writer identification. *Pattern Recognition Letters*, **28**(6), 719–727.

Séropian, Audrey. 2003. *Analyse de document et identification de scripteurs*. Ph.D. thesis, Université de Toulon.

Shahabi, F, & Rahmati, M. 2006. Comparison of Gabor-based features for writer identification of Farsi/Arabic handwriting. *Pages – of: Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

Siddiqi, Imran, & Vincent, Nicole. 2008. Combining global and local features for writer identification. *Pages – of: Proceedings of the 11. Int. Conference on Frontiers in Handwriting Recognition, Montreal*.

Siddiqi, Imran, & Vincent, Nicole. 2010. Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. *Pattern Recognition*, **43**(11), 3853–3865.

Siddiqi, Imran, & Vincent, Nicole. 2009. Combining contour based orientation and curvature features for writer recognition. *Pages 245–252 of: International Conference on Computer Analysis of Images and Patterns*. Springer.

Siddiqi, Imran Ahmed, & Vincent, Nicole. 2007. Writer identification in handwritten documents. *Pages 108–112 of: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 1. IEEE.

Slimane, Fouad, & Margner, Volker. 2014. A New Text-Independent GMM Writer Identification System Applied to Arabic Handwriting. *Pages 708–713 of: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE.

Sreeraj, M, & Idicula, Sumam Mary. 2011. A survey on writer identification schemes. *International Journal of Computer Applications*, **26**(2), 23–33.

Srihari, Bin Zhang Sargur N. 2003. Analysis of handwriting individuality using word features. *State University of New York at Buffalo, Buffalo, NY*, **14228**, –.

Srihari, Sargur N, Cha, Sung-Hyuk, Arora, Hina, & Lee, Sangjik. 2002. Individuality of handwriting. *Journal of forensic science*, **47**(4), 1–17.

Stewart, GW. 1998. *Matrix Algorithms. Volume I: Basic Decompositions. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.* Tech. rept.

Sun, Yi, Liang, Ding, Wang, Xiaogang, & Tang, Xiaoou. 2015. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, –.

Tahir, Muhammad Atif, Kittler, Josef, & Bouridane, Ahmed. 2015. Multi-label classification using stacked spectral kernel discriminant analysis. *Neurocomputing*, **171**, 127–137.

Tan, Xiaoyang, & Triggs, Bill. 2010. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing, IEEE Transactions on*, **19**(6), 1635–1650.

Tang, Youbao, Bu, Wei, & Wu, Xiangqian. 2014. Text-independent writer identification using improved structural features. *Pages 404–411 of: Chinese Conference on Biometric Recognition*. Springer.

Tao, Dacheng, Tang, Xiaoou, Li, Xuelong, & Wu, Xindong. 2006. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **28**(7), 1088–1099.

Tapiador, Marino, Gómez, Jaime, & Sigüenza, Juan A. 2004. Writer identification forensic system based on support vector machines with connected components. *Pages 625–632* *of: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer.

Tomai, Catalin I, Zhang, Bin, & Srihari, Sargur N. 2004. Discriminatory power of handwritten words for writer recognition. *Pages 638–641 of: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE.

Van Der Maaten, Laurens, & Postma, Eric O. 2005. Improving automatic writer identification. *Pages 260–266 of: BNAIC*.

Abdi, Mohamed Nidhal, & Khemakhem, Maher. 2015. A model-based approach to offline text-independent Arabic writer identification and verification. *Pattern Recognition*, **48**(5), 1890–1903.

Fiel, Stefan, & Sablatnig, Robert. 2013. Writer identification and writer retrieval using the fisher vector on visual vocabularies. *Pages 545–549 of: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE.

Helli, Behzad, & Moghaddam, Mohsen Ebrahimi. 2010. A text-independent Persian writer identification based on feature relation graph (FRG). *Pattern Recognition*, **43**(6), 2199–2209.

Siddiqi, Imran, & Vincent, Nicole. 2009. A set of chain code based features for writer recognition. *Pages 981–985 of: 2009 10th International Conference on Document Analysis and Recognition*. IEEE.

Walton, Judie. 1997. Handwriting changes due to aging and Parkinson's syndrome. *Forensic science international*, **88**(3), 197–214.

Wilkinson, R. Allen, Geist, Jon, Janet, Stanley, Grother, Patrick J., Burges, Christopher J. C., Creecy, Robert, Hammond, Bob, Hull, Jonathan J., Larsen, N. J., & Vogl, Thomas P. 1992. *The first census optical character recognition system conference*. Vol. 184. US Department of Commerce, National Institute of Standards and Technology.

Wu, Xiangqian, Tang, Youbao, & Bu, Wei. 2014. Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform. *Information Forensics and Security, IEEE Transactions on*, **9**(3), 526–536.

Xing, Linjie, & Qiao, Yu. 2016. Deepwriter: A multi-stream deep CNN for text-independent writer identification. *Pages 584–589 of: Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE.

Xiong, Yu-Jie, Wen, Ying, Wang, Patrick SP, & Lu, Yue. 2015. Text-independent writer identification using SIFT descriptor and contour-directional feature. *Pages 91–95 of: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE.

Yampolskiy, Roman V, & Govindaraju, Venu. 2008. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, **1**(1), 81–113.

Yang, Weixin, Jin, Lianwen, & Liu, Manfei. 2016. DeepWriterID: An End-to-End Online Text-Independent Writer Identification System. *IEEE Intelligent Systems*, **31**(2), 45–53.

Zhang, Honggang, Guo, Jun, Chen, Guang, & Li, Chunguang. 2009. HCL2000-A large-scale handwritten Chinese character database for handwritten character recognition. IEEE.

Zois, Elias N, & Anastassopoulos, Vassilis. 2000. Morphological waveform coding for writer identification. *Pattern Recognition*, **33**(3), 385–398.

Zuo, Long, Wang, Yunhong, & Tan, Tieniu. 2002. Personal handwriting identification based on pca. *Pages 766–771 of: Second International Conference on Image and Graphics*. International Society for Optics and Photonics.