# Automatic Handwriter Identification Using Advanced Machine Learning

**Amal Durou**

**Ph. D**

**February 2019**

# Automatic Handwriter Identification Using Advanced Machine Learning

**Amal Durou**

A thesis submitted in partial fulfilment

of the requirements of the

University of Northumbria at Newcastle

for the degree of

Doctor of Philosophy

Research undertaken in the Faculty of Engineering and

Environment

Department of Computer and Information

Sciences

# Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

I declare that the Word Count of this Thesis is approximately 35,514 words.

Name: Amal Durou

Signature: Durou

Date: 2019

# Acknowledgments

Completing this thesis successfully has been part of my aim and ambition of lifelong learning and studying for a PhD has been challenging but the rewards are immeasurable. I have broadened my horizons and I have gained a great deal of knowledge which will help me on both personal and professional levels. First and foremost, my deepest gratitude and thanks go to God for providing me with this chance and helping me to fulfil this mission. Without His generosity, none of this or any other achievement would have been possible. AL HAMDU LELLAH.

My special thanks go to my principal supervisor Prof. Ahmed Bouridane for providing me with his continuous support, inspiration, guidance and encouragement; and for giving me the opportunity to undertake the study under his supervision. My sincere thanks also go to Dr. Somaya Al-Maadeed (Qatar University) for her very useful and insightful discussions on the subject through Grant NPRP # NPRP NPRP7-442-1-082.

Last but not the least, I would like to thank my family: my husband, my parents and to my brothers and sisters; their supplication (Duaa) is the light that illuminates my path towards achieving my dreams, for supporting me spiritually throughout writing this thesis and my life in general.

# Abstract

Handwriter identification a challenging problem especially for forensic investigation. This topic has received significant attention from the research community and several handwriter identification systems were developed for various applications including forensic science, document analysis and investigation of the historical documents. This work is part of an investigation to develop new tools and methods for Arabic palaeography, which is is the study of handwritten material, particularly ancient manuscripts with missing writers, dates, and/or places. In particular, the main aim of this research project is to investigate and develop new techniques and algorithms for the classification and analysis of ancient handwritten documents to support palaeographic studies.

Three contributions were proposed in this research. The first is concerned with the development of a text line extraction algorithm on colour and greyscale historical manuscripts. The idea uses a modified bilateral filtering approach to adaptively smooth the images while still preserving the edges through a nonlinear combination of neighboring image values. The proposed algorithm aims to compute a median and a separating seam and has been validated to deal with both greyscale and colour historical documents using different datasets. The results obtained suggest that our proposed technique yields attractive results when compared against a few similar algorithms.

The second contribution proposes to deploy a combination of Oriented Basic Image features and the concept of graphemes codebook in order to improve the recognition performances. The proposed algorithm is capable to effectively extract the most distinguishing handwriter's patterns. The idea consists of judiciously combining a multiscale feature extraction with the concept of grapheme to allow for the extraction of several discriminating features such as handwriting curvature, direction, wrinkliness and various edge-based features. The technique was validated for identifying handwriters using both Arabic and English writings captured as scanned images using the IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting. The results obtained clearly demonstrate the effectiveness of the proposed method when compared against some similar techniques.

The third contribution is concerned with an offline handwriter identification approach based on the convolutional neural network technology. At the first stage, the Alex-Net architecture was employed to learn image features (handwritten scripts) and the features obtained from the fully connected layers of the model. Then, a Support vector machine classifier is deployed to classify the writing styles of the various handwriters. In this way, the test scripts can be classified by the CNN training model for further classification. The proposed approach was evaluated based on Arabic Historical datasets; Islamic Heritage Project (IHP) and Qatar National Library (QNL). The obtained results demonstrated that the proposed model achieved superior performances when compared to some similar methods.

# Publications

**Journal Papers: (Published)**

- A. Durou, I. Aref, S. Al-Maadeed, A. Bouridane and E Benkhelifa, "Writer Identification Approach Based on Bag of Words with OBI Features", Information Processing and Management, ISSN: 03064573, Elsevier. Volume 56, Issue 2, March 2019, Pages 354-366

**Conference Papers: (Published)**

- A. Durou, A. Bouridane, and S. Almaadeed, "Text Line Extraction on Color and Grayscale Historical Manuscripts using Bilateral Filter,".3rd International Conference on Signal, Image, Vision and their Applications (SIVA'15) University of Guelma, 2015, pp. 25–28.

- A. Durou, I. Aref, M. Elbendak, S. Al-Maadeed and A. Bouridane , "Improving the Performance of Offline Writer Identification using Dimensionality Reduction Techniques". Seventh IEEE International Conference on Emerging Security Technologies 6-8 September 2017, Canterbury, UK.

**Conference Papers: (Accepted and Presented in ICGS3 – Waiting for Publishing)**

- A. Durou, I. Aref, M. Elbendak, S. Al-Maadeed and A. Bouridane, "An experimental comparison of machine learning approaches for Writer identification in handwritten documents". 12th International Conference on Global Security, Safety & Sustainability (ICGS3). London, UK, 2019.

# Contents

# List of Figures

# List of Tables

# Acronyms

| Acronym | Description |
|---|---|
| ANN | Artificial Neural Network |
| BIFs | Basic Image Features |
| BoF | Bag-of-Features |
| BoVW | Bag-of-Visual Words |
| BoW | Bag-of-Words |
| CNNs | Convolutional Neural Networks |
| DBN | Deep Belief Nets |
| DIA | Document image analysis |
| DL | Deep Learning |
| DNN | deep neural network |
| DtG | derivative-of-Gaussian |
| GPU | Graphics Processing Unit |
| KNN | K-Nearest-Neighbour |
| KPCA | Kernel Principal Component Analysis |
| LCPH | local contour pattern histogram |
| LLE | locally linear embedding |
| MST | Minimal spanning tree |
| NN | Neural Networks |
| OBI | Oriented Basic Image |
| oBIF Columns | oriented Basic Image Feature Columns |

| **Acronym** | **Description** |
|---|---|
| PCA | Principal Component Analysis |
| RNN | Recurrent Neural Networks |
| SFH | stroke fragment histogram |
| SIFT | Scale Invariant Feature Transform |
| SOFM | Kohonen Self-Organising Feature Map |
| SRKDA | Spectral Regression Kernel Discriminant Analysis |
| SR-KDA | kernel discriminant analysis using spectral regression |
| SURF | Speed Up Robust Features |
| SVM | Support vector machine |

# Chapter 1

# Introduction

## 1.1 Overview

In recent years, there has been an increasing interest to use electronic documents over paper files, since they are inherently more secure than paper records while offering ease of retrieval and access. Despite this development, handwritten documents are still widely used by many companies and organisations because handwriting can be managed as a behavioural biometric issue. Several studies have revealed that different people have different handwriting styles, which means a handwritten document carries more information about the personality of the author that has written the document [1]. However, there is inherently a great degree of similarity in the writing styles of individuals, making it possible to distinguish the writer. Over the past few years, writer identification, which aims to determine the author of a specific document among different ones, has been an active research area and has recently been used in a large variety of real life applications such as data security in different services, financial activity, forensic science and can also be used in access control systems and the analysis of old documents [2],[3]. Although writer identification has been studied in the last few decades, it still remains one of the most important topics due to the recent development in the fields of pattern recognition and computer vision, which have led to a renewed interest in writer identification applications.

Handwriting is a process that has been mainly utilised for communication for centuries and its features have evolved in many ways over time. For instance, people start to learn handwriting skills at school depending on geographical location, and culture Figure (1.1). With the passage of time, unique writing characteristics have developed and progressed based on individual people and the parameters mentioned above. These individual writing characteristics can be used to recognise one person's writing style from another, thus making it possible to identify a writer.


(a) (b) (c)

Figure 1.1 Copybook styles1 (a) United States (Zaner Bloser) (b) Chile (c) German

Moreover, compared to electronic document analysis, a handwriting document contains more information about the personality of the writer. In addition, handwriting text has the ability to show important information about the author's status and can be utilised at some levels to indicate the physical, mental and emotional state of the author [3]. In a handwritten text, writing movement and stroke have been shown to identify the unique characteristics of the writer [4]. In some cases, it may be that two writers share some common attributes though their styles will be similar in their entirety. This illustrates that the writing style for each individual is consistent and unique and demonstrates how the writing styles of different writers will be diverse and unique.

Currently, there exist a number of historical studies that are focused on the investigation of ancient manuscripts and the experts, palaeographers, are responsible to perform the process of handwriting identification. As they are able to identify the writing features of a specific scribe and are able to distinguish the type of handwritten lettering used in a given historical period, they can predict the period of the writing. Therefore, developing a computer system to perform writing identification tasks of large amounts of historical documents will be a helpful application for the historians and end-users. Typically, ancient manuscripts can be accessed and collected from archives and libraries so that the collected information from individual documents such as punctuations and abbreviations can help palaeographers identify the writer and the date of creation of the manuscript Figure (1.2). A computer system can then be designed and implemented in order to enable the palaeographers to recognise the dating and classification of historical documents.

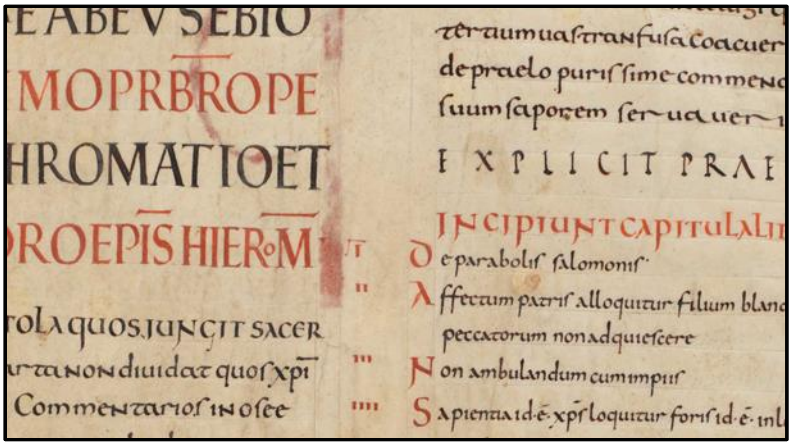


Figure 1.2 The evolution of writing from Roman square capitals through medieval Gothic to modern handwriting

## 1.2 Motivations

Automated handwriter identification is an approach used to assign a handwriting script sample to one author [1], [5] and it is currently a well renowned area in the field of pattern recognition. It begins by obtaining the biometric data of each writer to extract the unique and discriminative features of an individual. They are then matched against the stored features in order to get the most likely results. It has received profuse attention from the research communities and hence has become an active research topic[1], [2], [7], [8].

However, many issues are still unresolved such as the insufficiency of datasets and handwriting material in different languages. The recognition process is still an extremely challenging task due to the fact that each writer has personal handwriting patterns from different aspects such as qualities and styles that are subject to both inter-writer and intra-writer variations [6]. During the last few years, significant research efforts have been made to investigate issues associated with handwriting identification. During the last few years, the research activities surrounding handwriting identification have seen a renewed interest due to the possibility of developing numerous commercial applications including forensic science, document analysis and investigation of historical documents[1], [8],[9]. This thesis aims to investiage novel identification approaches especially for the identification of ancient manuscripts.

### 1.3 Aims and Objectives

The main aim of this work is to develop an accurate handwritten identification system by investigating new techniques and tools for the classification and analysis of ancient documents depending on different multi-scale features extraction techniques. In addition, another aim is to deploy the proposed techniques to support palaeographic studies by automatically deriving the writer of the relevant scripts. Therefore, the general objectives are to plan, analyse, design, build, and test novel classification algorithms and tools to support the palaeographic analysis of historical Arabic manuscripts. The objectives of this research can be summarised as follows:

- To investigate the concept of multi-scale feature extraction at a textural/global level for use in writer identification.
- To investiaget novel methods that can effectively combine different features I norder to improve identification performance.
- To investiage and develop a robust handwriter identification system that can be used by forensic analysts to analyse noisy and distorted documents.

## 1.4 Contributions of the Work

The key contributions of this work are:

1. **Text Line Extraction for Colour and Grayscale Historical Manuscripts:** A novel algorithm for the segmentation and detection of text lines of grey-scale and colour historical documents has been proposed to evaluate medial seams and separating seams. The novelty relates to an efficient method to separate the two seams using a bilateral filter approach [11].
2. **Writer Identification using the concept of Bag of Words with OBI Features:** Most existing handwriting identification systems use either statistical or model-based approaches. To further improve the performance, we propose to combine both approaches using Oriented Basic Image features and the concept of graphemes codebook. To reduce the resulting high dimensionality of the feature vectors a Kernel Principal Component Analysis has been used [12]. To further improve the identification performances, a number of nonlinear dimensionality reduction algorithms such as Kernel Principal Component Analysis (KPCA), Isomap, locally linear embedding (LLE), Hessian LLE and Laplacian Eigenmaps have been used and evaluated demonstrating that KPCA outperforms other techniques [13].

3. **A Comparative Study of Machine Learning Approaches for handwriter Identification:** A detailed comparative analysis approach to assess the performance of AlexNet deep learning architecture using a Support vector machine (SVM) and K-Nearest-Neighbour (KNN) has been carried out. The results have led to investigate the challenges of CNNs based methods for writer identification [14].
4. **Offline Handwriter Identification using Convolution Neural Networks:** A CNN architecture has been used as a feature extractor to evaluate the discriminative power of the features extracted by the connected layers of Alex-Net model. The features were evaluated separately and then combined before a nonlinear SVM is applied for classification to effectively improve the identification performances. The proposed approach was evaluated using different datasets including the Islamic Heritage Project (IHP), Qatar National Library (QNL) and Clusius datasets.

## 1.5 Thesis outline

The thesis is structured in eight chapters, a brief description of the contents of these chapters is given as follows:

- Chapter two gives some useful background and literature review including an overview of the main components that are required for any handwriter recognition system. The corresponding literature review relating to each contribution is also presented in the chapters.
- Chapter three discusses how to extract a text line on historical manuscripts (colour and grayscale) using a bilateral filter. An efficient method is proposed for the segmentation and detection of text lines of grey-scale and colour historical

documents. The algorithm emplys a bilateral filtering strategy to evaluate and separate two types: medial seams and separating seams.

- Chapter four proposes a new approach using a combination of statistical and model-based approaches using Oriented Basic Image features and the concept of graphemes codebook. To reduce the resulting high dimensionality of the feature vector, a Kernel Principal Component Analysis (KPCA) has been used. To gauge the effectiveness of the proposed method, performance analysis has been carried out using the IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting.
- Chapter five describes in detail how to measure and optimise the performance of an offline text writer identification system using dimensionality reduction techniques. A variety of nonlinear dimensionality reduction algorithms such as Kernel Principal Component Analysis (KPCA), Isomap, Locally linear embedding (LLE), Hessian LLE and Laplacian Eigenmaps have been used. The performance has been evaluated based on the IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting.
- Chapter six provides a comparative study of machine learning approaches for handwriter identification. It aims to assess and compare the performance of AlexNet deep learning algorithm with two machine learning classification approaches Support vector machine (SVM) and K-Nearest-Neighbour (KNN).
- Chapter seven presents an efficient approach for handwriter identification using deep learning technology by employing Alex-Net model to extract image features from the fully connected layers (FC6, FC7 or FC8) of the model providing 4096, 4096 and 1000 feature vectors, respectively. These feature vectors are then classified using an SVM model to effectively improve the identification

performances. The proposed approach is evaluated using different datasets including the Islamic Heritage Project (IHP), Qatar National Library (QNL) and Clusius datasets. In addition, experiments have been extended using IAM and ICFHR-2012 datasets to further demonstrate the efficacy of the proposed method. The results obtained have been compared against a few similar techniques and the analysis suggests that our method achieves superior results.

- Finally, Chapter eight highlights the conclusions of the thesis and lists several suggestions for future work.

# Chapter 2

# Background

A handwriter identification system allocates an unknown script to a specific writer from a group of possible writers. To achieve this, a one to many search procedure needs to be carried out on a large database containing handwriting samples of known writers to return a result as a list of candidates [1], [5]. The recognition process is still a very challenging task due to the fact that each person has different handwriting qualities and styles that are subject to both inter-writer and intra-writer variations [6]. During the last few years, significant research efforts have been devoted to tackling the problems associated with handwriting identification resulting in a plethora of papers, However, the topic still remains an active research area having many useful applications including forensic and historical document analysis [1]. Moreover, handwriting identification aims to simplify the task of forensic experts by providing them with semi-automated tools in order to enable them to narrow down the search to determine the final identification of an unknown handwritten sample. Therefore, an identification algorithm aims to produce a list of predicted writers of the unknown handwritten sample ranked in terms of confidence measure metrics for use by the forensic expert who will ultimately make the final decision.

These last few years have witnessed a steady rise in writer identification and verification [15]–[19]. Typically, the process to identify a writer based on a random handwritten sample can be a beneficial application and can be utilised in many different fields including forensic evidence and document analysis. This approach is different when compared to other biometric methods where there is a requirement to provide evidence of material and the details of the crime [20]. Moreover, various applications have been designed and developed based on writer identification and verification [20].

There are some examples from these applications such as how to recognise ink type [15], identify the handwriting in documents and recognise the language used [21], detect the forgery processes [22]. The technique has also been demonstrated for use to analyse medieval and historical documents [23]–[26] including to identify a writer from handwritten musical scores [27] and personal handwriting script recognisers [28].

Most of the early works on writer identification were developed mainly for the English language. Recently, the research community has targeted other languages, such as Chinese, Dutch, Arabic, and Greek resulting in a significant contribution to the field as a whole [3]. Although there are many researchers working in writer identification of the Arabic language, as far as research is concerned, there are still many issues surrounding this language that are active and under investigation.

## 2.1 Writing Process Development

In medieval times prior to the discovery of printing press devices, texts and documents would have been created solely by hand to write or copy text onto parchment. The tools used such as quill pen and ink would have, in most cases, been made by the authors themselves. At that time, literacy and writing where not skills carried by the majority. Generally, those who were literate would have been academics or professional writers with a view to producing literature such as legal letters, religious texts, medical and scientific documents.

Additionally, these writers would have the sole participants in almost all of the book production process. Usually, the parchment that would have been used as the writing material was made from animal hides made from sheep or goat skin. There would be a number of processing phases to carry out in order to produce this parchment to render them suitable for use as a writing material starting from cleaning, treating, washing, and stretching the skin.

In the last phase, the books would be put together by cutting the processed hides into booklet sized pages and then bound together to create the book in a form suitable for its purpose. Normally, there are two sides of the resulting parchment, the first side is called the inner 'flesh' while the other side is called the outer 'hair' and can shape part of the page texture and colour that is used in image reproductions.

The writing of these types of manuscripts is not an easy task and can be a lengthy process. Scribes observed that the period needed to complete an average manuscript is between 4-6 sides/day, which is equivalent to 24-40 sides/week [29]. The progress rate of the composition would be impacted directly with the selected font to achieve the target work, meaning that if the font used is more formal such as Textura then the produced work will contain on wonderful and higher-quality texts. Writing by using formal fonts required the scribe to lift the pen after each stroke. While the use of cursive secretary script is faster and is perfect for writing everyday documents, the font choice mainly depends on the text material and the language in which it was written [29].

One of the identification task issues is that professional scribes can create some fonts to assign to the text that they wish to copy, thus making the identification task more difficult. Furthermore, a group of scribes may work together on a single manuscript and attempt to change their personal writing styles in order to make their fonts similar. They would aim to make the writing transition process smooth and not easily detectable. From a writer identification point of view, this copying style is essentially a type of forgery and has been performed to present the manuscript in a professional manner.

## 2.2 Categorisation of Writer Identification Systems

Writer identification remains a challenging biometric recognition application. It is carried out as a pattern recognition problem to allocate an unknown written sample/pattern to one class (e.g., a writer) out of a set of classes (writers) [30]. The process can be defined as an algorithm/tool to assign a handwriting sample to one author/writer [1], [2].

### 2.2.1 Writer identification vs. Writer verification

As a general concept, writer recognition is an approach to detect the identity of a person using a computer system based on the analysis of his/her handwritten input such as paper documents. This is known as a behavioural feature. This recognition process can be illustrated as a biometric method and can be implemented in two different ways: writer identification mode or writer verification mode. In the first mode, the features of a group of known writers are extracted and stored in a database. Then, an unknown document is processed to identify the writer among the group. On the other hand, in the writer verification mode, an unknown document of a known writer is fed to the system to decide if both samples come from the same writer. This thesis is concerned with an investigation of writer identification mode [31]. A block diagram of both types has been illustrated in Figure (2.1).

### 2.2.2 Offline and Online modes

A typical writer identification system can operate in two methods: offline and online. The writing behaviour in the online mode is taken from the writer in real-time by converting it into a list of signals and directions via a writing tablet. However, for the offline mode, the scanned and digitised handwriting images are used to identify the writer [2], [32].

In more detail, in on-line mode, a written script is supplied by the user in real-time in order for the system to identify the writer (input text). This process can be done using a digitiser tablet with a special pen. A digitiser tablet (sometimes known as a graphics tablet) is a computer input device used to convert the input data (text) into a format suitable for computer processing at a constant rate [33]. In fact, there are some advantages to the on-line mode writer identification system over the off-line mode writer identification system. For example, the on-line mode is a real time system and does not need much work on the pre-processing stage [34]. The segmentation stage, as a pre-processing step, is x relatively easy to implement. Moreover, an on-line mode system can be used to process an individual’s signatures and the recognition process can be carried out using one-dimensional data. Conversely, on-line systems have some disadvantages including an inability to process text printed or handwritten documents requiring special tools.

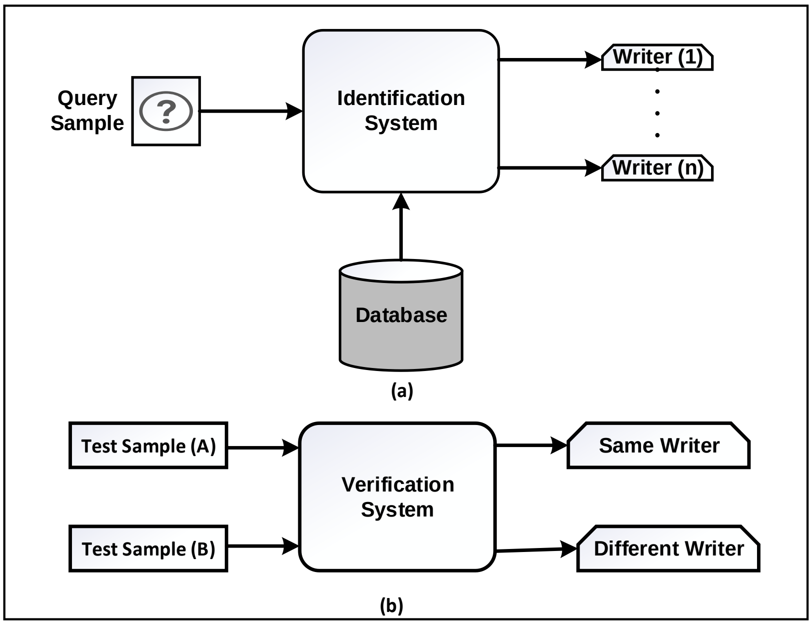



Figure 2.1 (a) Writer identification (b) Writer verification

On the other hand, an off-line mode system can be used to recognise handwritten captured as a two-dimensional image. In this case, an optical scanner or digital camera can be used as an input tool to capture the image of the handwritten text in a suitable format. The main advantage of off-line mode systems is that the identification task is carried out using stored or computer-based data, so such systems can be used for other applications such as processing historical and ancient documents. However, off-line systems have some disadvantages, for example, they do not operate in a real time mode and usually require significant pre-processing, feature extraction and classification [35].

### 2.2.3 Text-dependent and Text-independent handwriter recognition

Writer identification can be categorised into two approaches; text-dependent and text-independent [36]. In the text-dependent method, all writers provide the same known text [7], [10] while in text-independent counterpart, the dataset contains different text and sometimes different languages. The writer features can be captured using statistical parameters taken from different handwritten pages to generate a set of features that are insensitive to the text itself [26].

### 2.3 Components of handwriter recognition systems

A handwriter identification system consists of four main stages as shown in Figure (2.2): pre-processing, segmentation, feature extraction, and classification. It is worth noting that some of the previous works have used different approaches by merging some of these stages together, despite the fact that most of the works apply the same methodology [34].

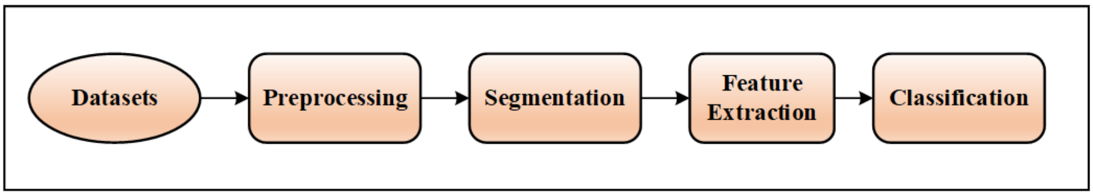



Figure 2.2 Block diagram of off-line writer identification system

#### 2.3.1 Pre-processing

The pre-processing stage is considered as the first step in any identification system. Sometimes, raw image data contains noise consequently deteriorating the performance of the identification thus rendering it unfeasible to process the input images. Hence, a pre-processing step is essential to remove irrelevant information (noise) to enhance the input image quality [19], [37]. To achieve this stage, a number of techniques have been proposed such as noise removal, thresholding, thinning, skew/slant correction, baseline estimation and normalisation [37] . The selection from these techniques should be based on the required recognition algorithm.

- **Noise removal**

Noise originates when input data (images) are created using scanners or writing tools. The effect of noise usually appears as a distortion in the image. Therefore, it is important to detect and remove the noise before starting any recognition process. There exist several approaches that can be used to remove these variations (noise) but, in particular, there are two that have been widely used in different previous works [37]; these are:

1) Filtering: filters have been designed and implemented to eliminate the noise and reduce various spurious effects so that the input image can be enhanced. Most of the filtering types are classified as spatial domain or frequency domain. There are many useful filtering operations such as sharpening, smoothing and adjusting brightness and contrast. One of the requirements is to design a linear spatial filter that can be used to modify an image by changing all the pixel values using a linear function of the values of nearby pixels. This filter is used to create a mask and then a convolution operation is applied between the image and the mask in equation (2.1).

$$Out(x,y) = \sum_i \sum_j w_{ij}\ inp(x-i, y-j) \qquad (2.1)$$

where $w_{ij}$ : pixel weight of the mask (grey level).

(i,j) : Pixel location inside the mask

Inp (x, y): The input image (original image).

2) Mathematical Morphological Operations: in image processing, this operation is a powerful tool used mainly to enhance the quality of the input images. It is very similar to the concept of convolution operation in the spatial filter, but in this approach, the

convolution process has been replaced by a logical operation. This operation uses the mathematical theory of depicting shapes using sets [38]. In the image processing field, the processes of erosion and dilation are used by mathematical morphology operation to examine the interaction between the target image and a structural element that has been selected. Actually, the operation of mathematical morphology is different compared to traditional linear image processing, because the mathematical morphology processes are non-linear, and therefore these processes are mainly depended on using another form of algebra that is not linear algebra [16].

- **Thresholding**

It is a simple technique used to analyse an image by partitioning it into a foreground and background. This method is performed using a threshold value. There are some advantages to this process as a less storage space is required and subsequently, the processing speed is faster. However, when we need to convert a grayscale image to a binary image, it is very important to select a suitable threshold value. The mapping function in equation (2.2) can be used as the simplest way to implement the thresholding operation [39].

$$out(x,y) = \begin{cases} 1 & if \quad Inp(x,y) \; < threshold \\ 0 & if \quad Inp(x,y) \; > threshold \end{cases} \qquad (2.2)$$

where:

*Inp (x,y):* The input image (original image).
*Out (x, y)*: the output image

There are two types of thresholding techniques. The first type, called Global thresholding, which can be applied when the intensity values of the input image are sufficiently distinguished into two groups (as objects and the background). This method aims to create a single threshold value that can be applied over the whole document. A histogram is utilised to compare the intensity values of the whole image and then the threshold value can be selected. Although, in some cases, the illumination level throughout the whole document is not uniform, and so, the use of global binarisation methods will act to generate marginal noise which means this method isn't suitable in these cases. Thus, to overcome this problem, local thresholding techniques have been proposed to solve this issue and it characterises the second type of thresholding technique [40]. The main idea of these techniques is to extrapolate a threshold value for each pixel based on its grayscale values of neighbouring pixels [41].

In the case of the images that have been affected with some distortion and noise e.g. poor contrast and/or complex patterns, there is a thresholding algorithm based on the selected feature can be used to binarise these images [42]. This algorithm has been established based on three steps; the first one is to set the thresholds using Otsu's algorithm [43]. Then, the texture features that have a relationship with each threshold are extracted using the run-length histogram. At the last step, the best threshold is chosen, and the desirable document texture features are maintained.

- **Baseline estimation**

Baseline estimation is an important phase in the writer identification task and sometimes called a reference line. It is an imaginary line but an essential task to extract the features of the input image. There are several baseline estimation techniques that have been proposed and centred around a baseline detection such as text normalisation, character segmentation, slant and slop correction (this has been adopted in this thesis for feature extraction). In addition, there are various potential problems that can arise and must be resolved in order to be able to extract a baseline. For example, in the case of Arabic scripts, a baseline detection of Arabic handwritten documentation as shown in Figure (2.3) is more difficult compared to other languages. Essentially, there are numerous techniques used to detect the baseline. One of the approved techniques for printed text is called the horisontal projection [38] .

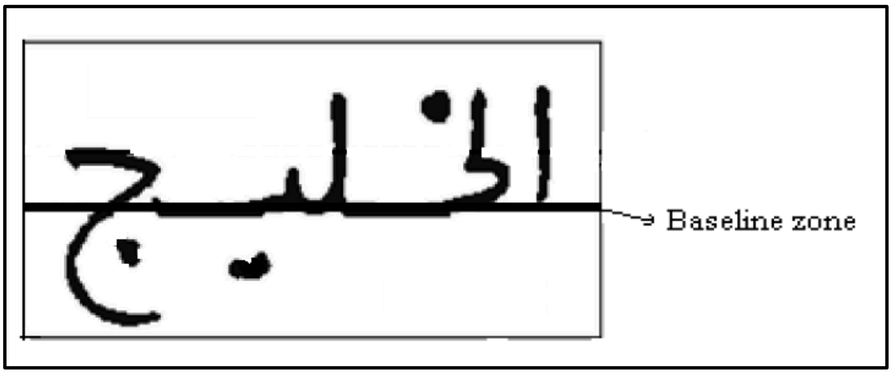



Figure 2.3 Baseline in Arabic handwritten

- **Normalisation**

In image processing, normalisation is a transformational process used to modify a specific range of pixel intensity values of certain images. It is an essential task in writer identification because the writing type is not the same style for all writers. One of the processes called size normalisation is employed to detect the character or word sizes and then modify them to a standard size [38]. Basically, there are various types of normalisation such as min-max normalisation, decimal scaling, and standard deviation method. Therefore, selecting a suitable normalisation method depends on the application at hand and the algorithm in which the normalised data will be used.

The min-max normalisation approach is a simple technique in which the data is fit in a pre-defined range as it is very common and usually more efficient. This is considered an important step in the identification stage in this work in order to address the issues relating to the differences in position and size, such as correlation and template matching. In essence, it can be stated that the main purpose of the normalisation process is to manage the variation of the text in the image and then produce a uniform output image. The variations of the handwritten text (from different writers) will then be reduced. Figure (2.4) illustrates a group of samples as an example that presents different stroke thickness and their normalised versions.

### 2.3.2 Segmentation

In the segmentation stage the output image is split into regions or categories. The newly defined regions usually correspond to a group of objects or some parts of those objects. The main idea is to allocate each pixel of the processed image to one of the newly defined regions. The input image illustrates a page of handwritten script or printed document [37]. Hence, the image segmentation process is mainly focused around subdividing the input image to homogeneous parts that are semantically meaningful based on certain properties like intensity or texture. In a writer identification task, the segmentation stage is an essential stage because the segmentation algorithm is used to divide an image that represents a document page into lines where each line is split into words and finally the words into characters [39]. Generally, the segmentation process can be classified into two types; external segmentation and internal segmentation [34].

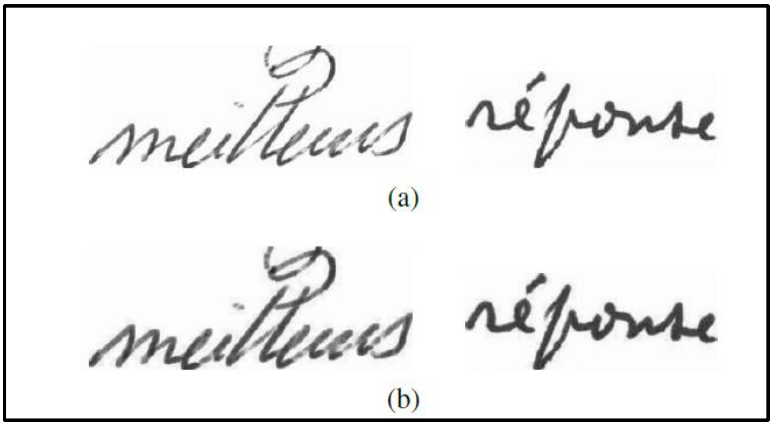



Figure 2.4 (a) Sample images (b) thickness normalised versions

External segmentation: this is used to handle page segmentation. As mentioned above, the document is divided into its basic parts, which are paragraphs, sentences, words or characters. It mainly addresses how to decompose the page layout into its logical parts [44]. Within this, there are two aspects that need consideration to accomplish the page layout analysis; structural and functional. A structural analysis deals with dividing the input document into the main components of the page such as paragraphs, lines, and words. A functional analysis looks into the basic principles used to identify the main components of the page such as title, abstract, etc. While internal segmentation is mainly used to split up the characters and letters that create words, but it can only be used with languages such as English that do not have a large amount of style variations. Therefore, it is very difficult to apply this method for Arabic texts.

- **Line Segmentation**

As the efficiency of segmentation has a direct impact on the performance of the recognition process, line segmentation must be carried out accurately and with a minimum number of errors. There exist various hand-written text line segmentation methods for extracting handwritten text such as the Hough transform [45]and horisontal projection[46]. Hough transform is a popular method for line segment detection and is used to detect broken lines, however, its main problem is that the processing speed is slow. While the other method, which is horisontal projection analysis, is known as the simplest approach that can be utilised to detect text lines. The local maxima and local minima of the projection profiles have to be identified [39].

- **Word Segmentation**

Word segmentation technique is typically used after the line segmentation technique and is used to divide the separated lines into words. The main idea of this technique is to start by detecting the connected components for each line in the target document. The connected components are then classified into several clusters by combining the word boundaries. In any case, finding the word boundaries within the text line is not a reasonably easy task. Therefore, in order to get high performance word segmentation, it is important to develop an effective algorithm for detecting word boundaries. One of the conventional approaches is to find the physical gaps between connected components [47]. This approach assumes that the size of the gap between any two words must be larger than the gap size between any two characters. This is can be applied for both English and Arabic [48].

On the other hand, the Arabic language has a different style of writing compared to the other languages such as the writing direction is from the right side to left and the words are created using connected characters. But in Arabic text recognition, the word segmentation process represents an important step to achieve the recognition process of the text in an efficient manner, because if it occurs any error during the segmentation process, then this will lead to produce a recognition error and reduce the overall performance of the process. Moreover, in the literature, there are two efficient methods that have been used before in some previous researches that are worked in Arabic text recognition. The first one is segmentation-based systems, while the other is segmentation free systems [49], a brief statement about both of them as following:

1- Segmentation based system: in this method, the segmentation process is classified into four phases [49]:
    a. Isolated characters method:  it is mainly used to recognise the isolated characters and numbers.
    b. Segmenting words into characters: This method tries to segment each word into individual characters, then apply the recognition process over each character individually.
    c. Segmenting words into their primitives: this method focuses to segment a sub-word or the connected components into symbols as shown in Figure (2.5(a)), the symbols represent a character, ligature, or part of a character.
    d. Integration of recognition and segmentation: this method aims to accomplish the recognition process as the first step, after that the segmentation can be performed as the second step.
2- Segmentation free system: Segmentation free system: as illustrated in the figure (2.5(b)), this method focuses to carry out the recognition process over the whole word without the segmentation process.

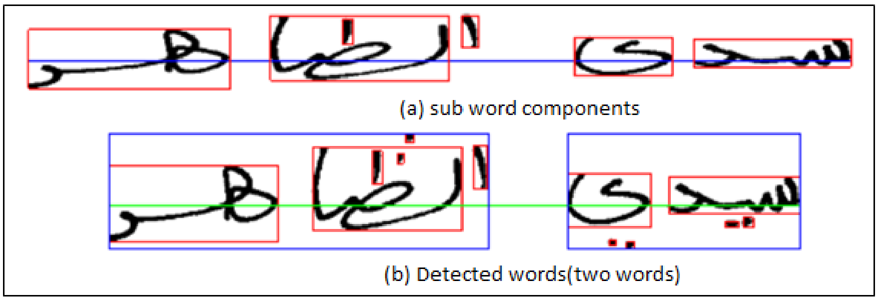



Figure 2.5 (a) sub word components, (b) detected words [37]

### 2.3.3 Feature Extraction

Feature extraction is one of the most important stages of a pattern recognition system as it is a key process to identify and represent the different objects in the image. This process is also essential for representing the objects by using a set of features that are created as a form of a vector called a feature vector. The main purpose of this vector is to represent the objects and then at the last stage, to classify these as will be explained in the next section.

Feature extraction has been used for a long time and previous research has targeted feature extraction algorithms that were proposed and developed based on different parameters such as colour, texture, and shape. Its aim is to convert the target image into a group of features that can be used to represent that image, which means, the arbitrary data (input image) is treated and broken down to qualitative characteristics or numerical features [50].

Usually, each image has various contents and elements that are used to produce the full image such as the colour, texture, shape, etc. One of the critical procedures in the feature extraction process is a feature selection that has proven its effectiveness in many of the previous applications. In spite of various methods that have been suggested to extract features in literature, it is still very difficult to select a suitable feature that is expected to produce and guarantee a high-performance system. Colour is the clearest visual feature and colour moments have been used for indexing and recovering images. Colour moments comprise three parts; the first order is called the mean, the second order is standard deviation and the third order is skewness and all of them have shown efficacy in representing colour distribution of images[51]. Stricker and Dimai [52] have proposed a method whereby subdividing images and relying on fully segmented images is the main approach. Their approach has been developed based on 5 partially overlapping, fuzzy regions. In addition, the texture is known as a powerful cue in region-based image segmentation techniques. One of the most important features in various pattern recognition is a texture feature, which is known as a low level feature. It is mainly used to describe a whole image or a region [53]. Texture analysis has long been used in various research and there are many algorithms, e.g. using Gabor wavelet features and then comparing with multiresolution texture features [54]. Due to the similarity between multi-resolution filtering methods and the process of human visual systems, the authors in [6] have proposed a method for using Gabor and Wavelet Transform techniques for texture description by using the analysis of spatial frequency content.

There are several types of features that can be examined and investigated when processing these images. The numerous features that can be examined and then employed to distinguish the main markers are general features and domain-specific features.

- **General features:** these types of features can be classified as independent features such as colour, texture, shape and spatial. These features are divided into:

1. Pixel-level features: Features can be determined at the pixel level, e.g. colour, location.
2. Local features: These types of features can be calculated across a part of the image. The specific properties within the local regions of the image are considered such as edges, lines, curves, etc [55], such as:
   - ✓ **Local Binary Pattern (LBP):** this is a widely used tool in image processing and computer vision fields. As a non-parametric approach, LBP represents efficiently local structures of images by comparing each pixel with its neighbouring pixels. Tolerance is considered as one of the most important properties in the LBP approach [56].
   - ✓ **Histograms of Oriented Gradient (HOG):** this is a feature descriptor widely used across multiple domains in order to characterise objects through their shapes. In most instances, local object appearance and shape can be depicted by the distribution of local intensity gradients or edge directions [57].
   - ✓ **Speeded Up Robust Features (SURF):** SURF is a feature detector and descriptor algorithm and has been classified as an effective approach in object detection and pattern recognition applications. If applied for real-time applications, the method would require a large computational complexity, which is considered as a major disadvantage of this algorithm [58]. On the other hand, due to its powerful attributes such as scale invariance, translation invariance, lighting invariance, contrast invariance, and rotation invariance, the

algorithm can be used as a very effective approach in several applications.

- ✓ **Scale Invariant Feature Transform (SIFT):** SIFT algorithm is effectively used to resolve affine transformations, intensity, and viewpoint change in matching features. It has been designed based on four main steps. The first one is to evaluate a scale-space extremum by using the Difference of Gaussian (DoG). The second step is a key point localisation where the key point candidates can be localised and refined by excluding the low contrast points. Thirdly, a key point orientation is allocated based on the local image gradient and the last step is to use a descriptor generator in order to calculate the local image descriptor for each key point based on image gradient magnitude and orientation [58].

3. Global features: Here the features are calculated over the entire image, which means focused over the whole image such as intensity histogram, contour representations, shape descriptors, and texture feature, etc.

- **Domain-Specific Features:** An application of dependent features such as human faces, fingerprints, character recognition and conceptual features.

Customarily, the extracted features can be divided into two types. The first type relates to low-level features where the features are extracted from the original images, while the second type is called high-level features that mainly depend on low level features. As previously mentioned, the extracted features are used to create a feature vector to recognise and classify the different objects inside the processed images. A brief explanation about these features is as follows;

- **Texture:** is one of the significant properties of an image. It is a robust regional descriptor and can be classified as a low-level feature. Usually, it is used for the retrieval problems. It can be used together with the colour feature to depict the contents of the images since the colour feature solely doesn't have the capability to identify the image. There are several ways to describe the main characteristics of textures such as coarseness and regularity. There are also many ways to detect texture features such as Gabor Filter. The authors in [59] clarify that the texture is a duplicated style or model of information, which means, an organisation of structure within a constant duration. Usually, texture indicates surface characteristics and also the look of an object from different sides such as size, shape, density, arrangement, the ratio of its primary parts. The fundamental step that needed in order to collect such characteristics is called texture feature extraction, which is mainly a texture analysis process. As a result of the importance of texture information, so the texture feature extraction approach is considered essential in several image processing and computer vision applications such as remote sensing and medical image [50].
- **Shape:** is considered as an important visual and primitive feature and is normally used to describe the image content. However, the description process of the shape content has faced some problems with regard to determining the content because the comparison to find the similarity between shapes is not an easy task [60]. This leads to identifying two fundamental steps in order to retrieve and restore the image based on the shape feature. These steps are to extract the features and then to try to detect the similarity between the extracted features. In fact, there are two

techniques that can be used for shape-based feature extraction; the first one is region-based, here the shape is described using the entire region of an object. The second technique is contour based the utilisation of local features as boundary segments.

### 2.3.4 Classification

The main task of the classification phase is to allocate a sample (or a form) to a predefined class based on a relation between the data processed objects with a pre-defined class label [61]. Usually, the classification methods are essential during the learning phase because it allows the target model to learn over the training set and then each sample of the test data can be classified in an effective manner and assigned into one of the predefined classes [37]. There are many types of classification algorithms such as K-Nearest Neighbours (KNN), Neural Networks (NN) and Support Vector Machine (SVM) [62]. In the next sections, we discuss briefly each type.

- **Artificial Neural networks (ANNs):** ANNs are a computing architecture formed with a set of adaptive neurons that are connected in parallel having the ability to adapt to the changing of processed data and learn the characteristic of the input pattern. There are a lot of applications that have been developed based on this concept such as text recognition and also can be used for classification. Moreover, NNs are classified as a nonlinear system and are identified by a specific network topology that is determined by the characteristic of the neurons and the learning methodology.
- **K-NN Classifier:** It is a supervised machine learning algorithm that can be used mainly for classification and regression in pattern recognition fields. It aims to classify the objects based on the closest training examples within the feature space.

One of its advantages is its simple structure I nwhich the key idea is to classify the patterns based on a majority vote of its neighbours. This means the patterns are allocated to the most common class between several classes that defined based on its K nearest neighbours. The training data are a dataset of examples used for training. It has been created as vectors attached with a label for each vector to define to which class it will belong. In the first stage of the learning process, which is the training stage, the algorithm reads the feature vectors and then stores them with their class labels. After that in the classification stage, the query vector, which is a test sample (unknown sample) will be classified by allocating the most appropriate label from the labels defined in the training dataset. In this method, a distance metric is used to measure the nearest neighbours for the sample test compared to the samples defined in the training dataset [63].

- **Support Vector Machine (SVM):** It is a supervised learning model useful for classification problems and regression analysis applications. Moreover, SVM distinguishes as a binary linear classifier that can provide a spatial illustration of the training dataset. Usually, it set up a model that can be used to map new data within the same space for classification. The main idea of this procedure is to get the best hyperplane allowing for the the split the data into two classes. A hyperplane is a linear classifier that can be constructed in a high- or infinite-dimensional space of learning patterns. Figure (2.6) shown the hyperplane in two dimensional and the support vectors are represented by all the points located close to the hyperplane.

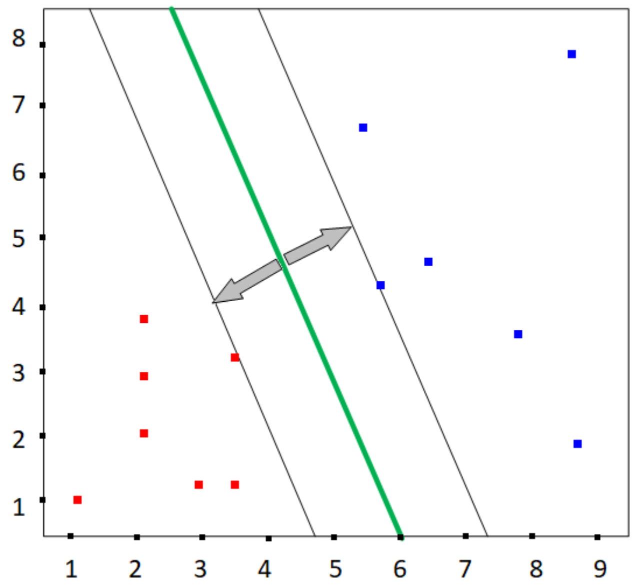


Figure 2.6 SVM example. Data points of a two-dimensional classification problem separated by a hyperplane.

Moreover, there is a space located between the hyperplane and the support vector which is called the margin. If we need to classify new data, so one of the hyperplane sides will be selected to map the target data. Usually, there are differs hyperplane, that can be used to split the training dataset. The optimal hyperplane is located as far as possible from the locations of the data points [64].

- **Euclidean Distance:** is also known as Euclidean metric and distance magnitude. Its essence is to measure the exact straight-line distance between any two points in the Euclidean space [65]. The points are located on one, two or three dimensions. Figure (2.7) illustrates how to measure the distance of three individual points represented by two variables (two dimensions), variable-1 represents the x-axis, while variable-2 is used for the y-axis.

  As shown in Figure (2.7), the Euclidean distance has been represented by the line between any two points, i.e. every two persons. In this case, there are three distances that need to be measured, one for each distance between two persons. Equation (2.3) is used to evaluate the distance between each two points [65].

$$d = \sqrt{\sum_{i=1}^{v}(P_{1i} - P_{2i})^2} \qquad (2.3)$$

Where, v: number of variables (in the figure (2.7), v=2).

This equation is used to calculate the difference between two points where the results are squared and finally used to calculate the summation for v variables. In the case of Figure (2.7), three distances would be calculated, which are: ($P_1$–$P_2$), ($P_1$ – $P_3$), and ($P_2$ - $P_3$).

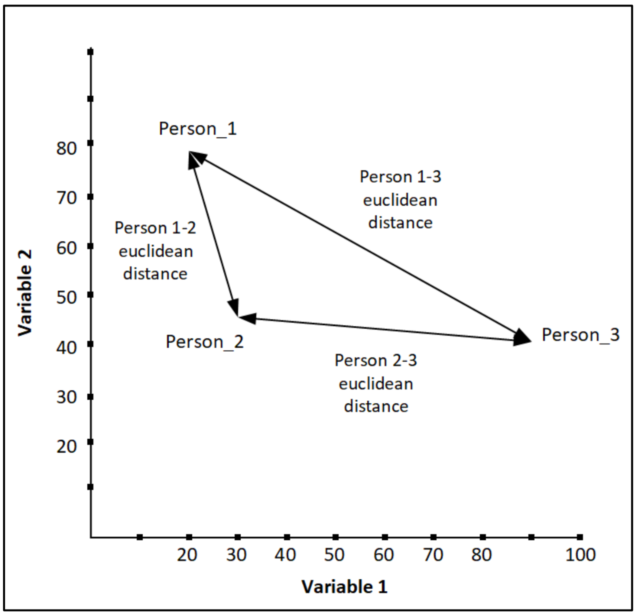



Figure 2.7 Measure the distance on two dimensions

Equation (2.4) has been used to find the exact distance between the two variables [65]. In Figure (2.7), there are three persons (points) and the equation is used to calculate the difference between every two variables and the result would be squared and finally used to find the summation for p persons (points).

$$d = \sqrt{\sum_{i=1}^{p}(V_{1i} - V_{2i})^2} \qquad (2.4)$$

- **Chi-square Distances:** The chi-squared distance d(x,y) is a distance between two histograms x=[$x_1$,..,$x_n$] and y=[$y_1$,...,$y_n$]. Both histograms are normalised, i.e. their entries sum up to one. The distance measured is usually defined in equation (2.5) as follows:

$$d(x,y) = \sum \frac{(x_i - y_i)^2}{(x_i + y_i)} \qquad (2.5)$$

  This measure is often used in computer vision to compute distances between some bag-of-visual-word representations of images.

  It is a simple method and depends on the normal distribution. The main step of the Chi-squared method is to calculate the square of the distance to the mean. After that, we need to check, if the underlying observations are normally distributed, so we get the chi-squared distribution [66].

- **Manhattan Distance:** It is a distance measure used to calculate the distance between any two points along the axes at right angles. In the case of two vectors, the distance is calculated as equation (2.6).

$$d = \sum_{i=1} |x_i - y_i| \qquad (2.6)$$

  for all the dimensions of the vectors [67]. For example, if we have two points such as $P1(x_1, y_1)$ and $P2(x_2, y_2)$, the distance is $|x_1 - x_2| + |y_1 - y_2|$. This means, calculate the differences of the points coordinates and take the absolute values and finally find the summation [67].

### 2.4 Datasets Used

In this thesis, all the experiments have been conducted using five available datasets in order to evaluate the performances of the proposed algorithms. Our datasets comprise of four Arabic; ICFHR 2012, Islamic Heritage Project (IHP) and Qatar National Library (QNL), one is English: IAM and one is Clausius Dataset. In addition, in this research, all the experiments have been conducted using the same training and test set samples. The details about the datasets have been provided in the next sections.

### 2.4.1 IAM Dataset

The IAM English handwritten dataset [68] is one of the most widely used datasets for the evaluation and validation of handwrite identification and verification systems. The dataset comprises of 1539 English handwriting documents generated from 657 writers and saved as digital images having a resolution of 300 dpi. To ensure a fair evaluation of the proposed technique a similar environment, as used by [69]and [70], has been maintained and considered in this chapter. Therefore, for the testing phase, the IAM dataset comprises of a total of 1314 handwritten samples with two samples per writer. On the other hand, the training process consists of the third and the fourth samples from the 127 writers who provided more than four samples. The data that has been used for the testing process is gathered from the first and second samples of all the 657 writers. In addition, care was taken in order that the training part and the testing part of the data are separated. Figure (2.8) illustrates some samples from the IAM dataset.

### 2.4.2 ICFHR-2012 Dataset

ICFHR 2012 dataset for the Arabic language is a large dataset [71]. The documents have been digitised and saved as greyscale PNG images having various text contents. In this dataset, more than two hundred writers were asked to write three different content Arabic paragraphs. The pages have been divided into a training set and a testing set. The first two paragraphs from each page/writer are segmented as a training set while the third paragraph is stored in the testing set. Figure (2.9) illustrates some samples taken from ICFHR 2012 dataset.

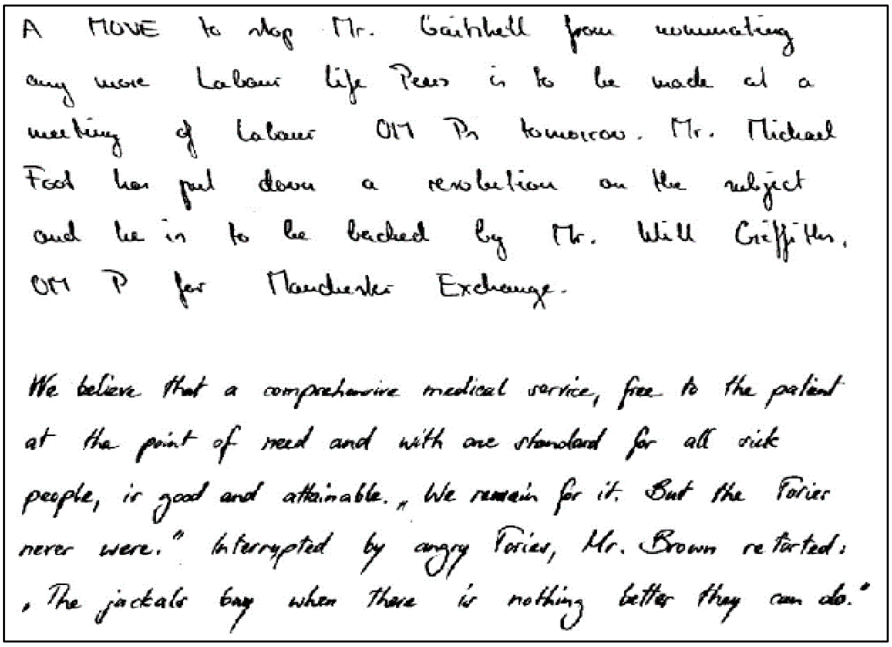


Figure 2.8 Two samples from IAM dataset.

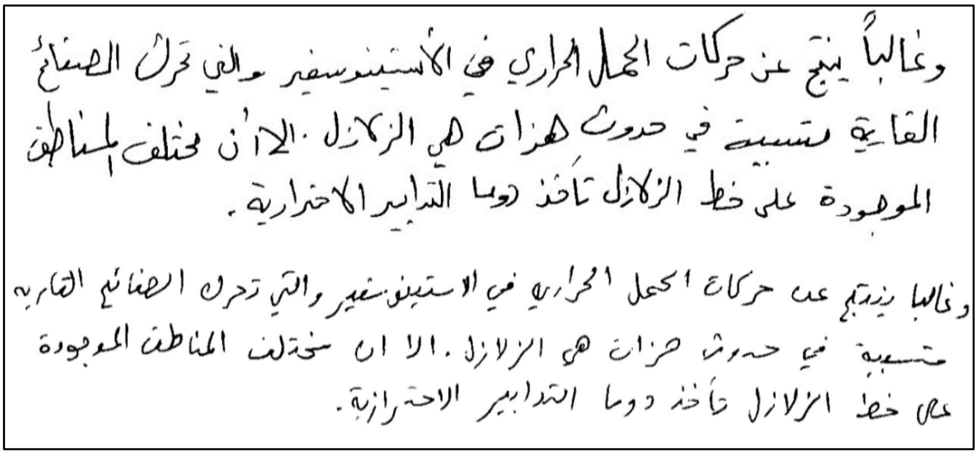


Figure 2.9 Two samples from ICFHR 2012 dataset

### 2.4.3 Islamic Heritage Project (IHP)

Manuscripts were collected from the Islamic Heritage Project (IHP) [72] which has been created as the first historical Arabic data set to be used for writer identification. The manuscripts were written during the period from the 13th century to the early 20th century. Figure (2.10) illustrates some example scripts from the dataset. Most of the manuscripts are written in the Arabic language, except for two written in the Ottoman language using the Arabic alphabet. The dataset includes works by 29 writers, each contributing 10 or 11 scripts, and was saved as 300 dpi digital images.

Figure 2.10 Example pages of IHP

### 2.4.4 Qatar National Library (QNL)

The QNL dataset [73] is produced by the Qatar National Library and contains a rich archival collection of historical Arab manuscripts. This library has digitised more than 5000 documents during the last few years. The QNL dataset has been built from real and old handwritten scripts with different backgrounds and written by different authors. It was mainly created to train Arabic handwriting identification systems and has since been widely used to evaluate such systems. The dataset used in this work contains 1210 documents collected from 121 different writers. The manuscripts in the QNL dataset have been scanned at 300 dpi resolution and saved in “png” image format. Figure (2.11) shows some samples from the QNL dataset.

Figure 2.11 Example pages of QNL.

### 2.4.5 Clusius Dataset

The Clusius dataset contains 1600 forms with a resolution of 300 dpi. The documents were created by Carolus Clusius (1526-1609), who was one of the most important botanists and a group of persons from different fields such as politics, religion, gardening and travel. This dataset, which is provided by the Huygens Institute for History in the Netherlands, has also produced an electronic-copy of the correspondence of Carolus Clusius using the collaborative editing tool eLaborate [74]. During the last two centuries, these historical forms have gained great attention from historians and biologists, but unfortunately a complete edition of the correspondence has not been created until now. Figure (2.12) shows some samples from the Clusius Dataset.

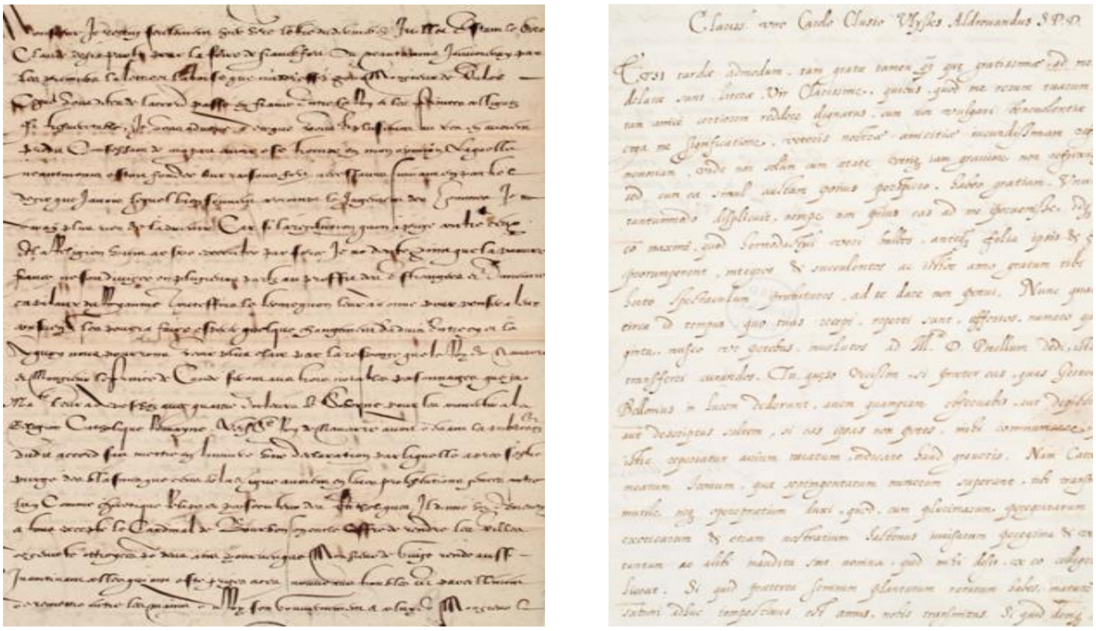


Figure 2.12 Sample images of Clusius Dataset

### 2.5 Summary

This chapter has reviewed previous state of the art systems of writer identification. In the literature review, the components of handwriter recognition systems have been discussed in detail. We list all the datasets that have been used to evaluate the proposed approaches.

The next chapter presents a line extraction algorithm that is designed in order to evaluate two types: medial seams and separating seams. The novelty relates to an efficient method to separate the two seams using a bilateral filter approach.

# Chapter 3

# Text Line Extraction using Bilateral Filter

### 3.1 Introduction

Text line extraction is one of the important layout analysis steps in document image analysis (DIA) and it is an essential part of script identification technique from handwritten document images. In this work, we have developed an algorithm that analyses the grey-scale and colour historical documents by using seam carving method that typically includes two stages with the first one being medial seams while the other is separating seams [75]. The main step of this algorithm is to calculate separating seams between two consecutive text lines without cutting through line components. Seam carving is a suitable algorithm for this application because it computes the minimum energy seams of the document image. The algorithm can recognise text components as those having high-energy regions while the paper or parchment background has low-energy regions.

Moreover, when Gaussian filtering is used in uniform areas of the image, it does not make the filtering across object borders. On the other hand, using a bilateral filtering approach will output a more pleasant result, because it avoids the introduction of blur between objects while still removing noise in uniform areas [76].

Therefore, the main goal of this work is to develop a modified text line extraction algorithm using the method implemented before in [77] which was based on Gaussian filtering. We propose to use a bilateral filter to efficiently separate the two types of seams based. To the best of our knowledge, it is the first time that a bilateral Filter is investigated for text line segmentation of handwritten documents. Most of the previous works construct a text line extraction algorithms based on Gaussian filter such as [75], [77], [78]. The main idea of the bilateral filtering approach is based on smooth images while still preserving the edges through a nonlinear combination of neighbouring image values.

In the rest of this chapter, we describe our developed algorithm by using a bilateral filter, present the benchmark dataset and report experimental results. Finally concluding remarks are given in the last section.

## 3.2 Text Line Extraction Approaches

During the last years, the main task for most of the document processing applications is to process handwritten document images and trying to extract unconstrained text lines by different efficient methods. It has received a great deal of attention over the past few decades. According to previous works, the process of the text line extraction can be divided into three main approaches; top-down, bottom-up, and hybrid [78] The following sections will describe each approach in more detail.

### 3.2.1 Top down methods

In Top-down approaches, the document image has been iteratively divided into several blocks to separate the required part. These approaches are mainly depended on previous information on the documents such as inter-line / inter column space distances, or they can be used as a document model to split up the documents into blocks and then the blocks into lines.

- **Projection based methods:** In these methods, the projection histogram profile has been analysed and used to locate the white separations. The projection angle has to be adjusted very well against the document direction, then we can observe a big difference between two successive values in the histogram and this can be considered as a border between two blocks of text, but here the main issue is the evaluation of the differences in the histogram. It will be more helpful, if we use prior information about the target document such as the number of columns and margins, then the separations process can be located very easily [78], [79]. In the literature, there are previous works that have used the projection profile as an approach for the text line extraction process. However, this method cannot guarantee giving good results when the space between the lines is not consistent. Moreover, there are two items; Text line skews variability and touching, that impact directly the performance of this method. Therefore, the projection profile method has been changed to be used in an adaptive manner, which means, it will not process the entire text line, but adaptive fractions will be employed to calculate the projection profile [78], [79].

- **Document model-based methods:** In varies fields, the priori information is highly organised and there are numerous segmentation issues are related to the density of information that has not been solved by using traditional approaches that existing in the literature [78]. A model-based method has been proposed in order to find broken text lines in noisy documents. The directional single-connected chain and vectorisation-based algorithms have been used to extract the line segments. After that the line model with three parameters will be created; the skew angle, the vertical line gap, and the vertical translation. Then based on this model, the high-level contextual information can be integrated to enhance detection results [79].

### 3.2.2 Bottom-up methods

Usually, to process handwritten documents, most of the line extraction methods have been developed as a bottom-up class, because they can be used to address noise issues and writing variation in an efficient manner. Bottom-up methods mainly deal with low-level elements such as pixels. The connected component-based approaches represent the core of these methods. They are collected into large compounds such as words, lines, and blocks. During the research process, the basic rules, which are utilised in a different way, were derived to create geometric relationships between adjacent blocks, such as distance, overlap, and size compatibility. The main difference between the works mainly focusses on whether each work can overcome the differences in the space and the effect of the script and the writer's characteristics. In the literature, various methods have been presented and used for clustering connected components such as KNN, Hough transforms, Smoothing, Repulsive-attractive Network and Minimal spanning tree, Each method is described separately in the next sections [78], [79].

- **KNN:** This approach aims to classify the components based on some basic geometrical rules among k-nearest neighbors. The authors in [80] presented a graph-based approach that has been applied to extract text lines from Arabic documents. The first step is to calculate the local orientation of each main component in order to construct a similarity graph. After that, they calculate the similarities between non-neighboring components using the shortest path algorithm. Then they use the graph to obtain the text lines by applying two estimates based on the Affinity propagation and Breadth-first search.
- **Hough transform:** Hough transform is another method used to extract text lines in bottom-up approaches. It is mainly related to the voting points that have been

used to represent the text lines. The authors in [45] presented an approach that used a block-based Hough transform and was used to detect the potential text lines. In this approach, there is an essential step, which is the post processing phase that can be used to correct possible false alarms. The block sizes can be evaluated by calculating the average of the character sizes in the document.

- **Smoothing:** The smoothing technique that is called Run Length Smoothing or RLS, is mainly used to remove the small spaces that are located between any consecutive black pixels. This process is done in horizontal direction so that it will produce connected pixels [45]. On the other hand, the boxes that contain the successively connected components inside the image will be used to form the lines. The approach in [81] , based on RLSA can be used to extract the lines in a grayscale image. In this approach, the image gradient is extended horizontally, and tilt angle can be added within the range of $30^o$. The main advantage of these approaches can be used in different directions if needed.
- **Repulsive-attractive network:** This approach is a dynamic system and can be used to reduce energy on a textual image by using attractive and repulsive forces that are located on the components of the network and the document image. The results of the experiments have shown that the network can be used to extract the baselines under the noise environment and interferes between the ascending and descending pieces of characters in two lines [79].
- **Minimal spanning tree (MST):** If we consider the connected components in the target document as the vertices of a graph, then we can get a complete undirected graph. For some cases, a tree is called a spanning tree of a connected graph that can be known as a maximal acyclic set of edges or for other cases as a tree that contains every vertex [78] .A minimum spanning tree (MST) of a graph or the

minimum weight spanning tree is the number of edges that equals the minimum possible total edge weight compared to all the spanning trees of this graph. Kruskal algorithm is usually used to produce a minimal spanning tree of a graph. The main idea of this algorithm is to construct the tree by including the remaining unused edge with the low cost and need to make sure that all the vertices are connected [79].

### 3.2.3 Hybrid methods

These methods aim to merge top-down approaches with bottom-up approaches in order to improve the results. In this class, the approaches have been found based on the deformable contour model, which is an analytical method that can perform interactively on the modeling. It permits to make the change to the representation of the model across the solution of the minimisation problem introduced in the modeling. This change has been managed in two directions; time and space [79]. The authors in [82] presented a novel hybrid approach that develops the strengths of the bottom-up and top down approaches and on the other hand, tries to reduce their weaknesses as much as they can. The proposed approach is aimed to overcome strongly issues that usually faced in historical documents such as skew, arbitrary warping, touching components and the presented of different types of noise. This method has been implemented and successfully validated.

## 3.3 Proposed System

Our proposed method consists of two stages. In the first stage, we compute the medial seam using a projection profile matching approach similar to [83]. After that, the separating seam is computed using the bilateral filtering of the seam carving procedure. Many state-of-the-art approaches have been proposed to tackle these challenges using techniques, such as multiresolution transforms, nonlinear filters, nonlinear stochastic regularisation. Here, we have used a convention that indicates an image

I $\in R^{n\times m}$ that comprises of n rows and m columns. The notation $\boldsymbol{I_{i,j}}$ indicates the image value at the location i-th (row) and j-th (column). The origin of the coordinate system is located at the top left corner of the image. In the next sections, the medial seam and the separating seam are described in more detail.

### 3.3.1 Medial Seam

Our medial seam computation method is inspired by the projection profile matching approach of [83]. A projection profile is a histogram of the number of black pixel values accumulated along parallel lines taken through the document as shown in Figure (3.1). In the literature of document image analysis, projection profile methods have been used for skew estimation [84], text line segmentation [85], and page layout segmentation [86] and in many more applications. For an uncompressed document of '**m**' rows and '**n**' columns, the page is divided into a number of slices **r**, the width of each slice equal **(w=m/r)**. Subsequently, we can apply the Sobel operator to I in order to calculate its edge image S$\in R^{n\times m}$. Then we can compute smoothed horizontal projection profiles $\boldsymbol{P_g^c}$ of **S** for each slice independently as indicated in equation (3.1):

$$\boldsymbol{P_i^c} = \sum_{\boldsymbol{j=k}}^{\boldsymbol{k-w-1}} \boldsymbol{S_{i,j}} \qquad ,\boldsymbol{P^c} = \{\boldsymbol{P_i^c}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}} \;, \qquad \boldsymbol{P_g^c} = \boldsymbol{g}(\boldsymbol{P^c})$$

$$c=1,\ldots\ldots,r\;,\; k\in\{1,1+w,\ldots\ldots,1+(r-1)w\} \qquad (3.1)$$

Where: **g** - cubic spline smoothing filter.

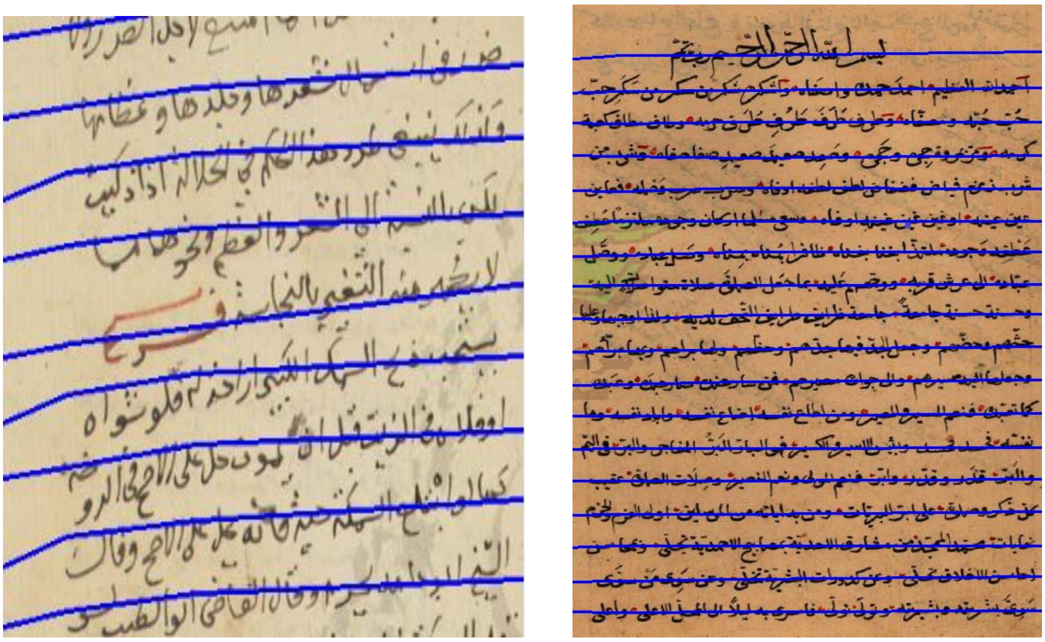


Figure 3.1 Example of computed medial seams (blue lines)

### 3.3.2 Separating Seam Computation using bilateral filtering

The filtering process is considered one of the most important operations in image processing. Bilateral filtering is one of the filters types and mainly used for smoothing images while preserving the edges. This can be done through a nonlinear combination of adjacent pixels values. Moreover, bilateral filtering is designed based on a combination of a domain and range filtering [87]. Domain filtering is used to measure the geometric closeness from the neighborhood center to the nearby points. Therefore, domain filtering can be seen as Gaussian filtering while range filtering is used to measure and detect the similarity between two pixels. then the values of the similar pixel are weighted and marked as a high influence and rejecting dissimilar ones [88].

The separating seams define the upper and lower boundaries of text lines as shown in Figure (3.3); i.e., determine the text strip, which is necessary to assign in-between component to the right text lines and accurately determine the pixels that need to be updated in the seam-map to avoid re-computing the seam-map after each line extraction. The separating seams of a text line are generated with respect to the medial seam of their text line. We adapt the seam carving algorithm proposed in [75] to compute the separating seams. We include the regional constraints of the computed medial seams and modify the seam computation so that it can handle non-rectangular image regions. The energy map is the derivative image (**I**) of the greyscale manuscript page using Equation (3.2) below:

$$E_{i,j} = \left|\frac{I_{i,j+1}^{\delta_S} - I_{i,j-1}^{\delta_S}}{2}\right| + \left|\frac{I_{i+1,j}^{\delta_S} - I_{i-1,j}^{\delta_S}}{2}\right| \qquad (3.2)$$

In addition, to the best of our knowledge, no previous studies have used bilateral filtering in order to compute the separating seam. The bilateral filter is considered as a non-linear, edge-preserving, and can be used as a noise-reducing smoothing filter for images. The main idea is to replace the intensity of each pixel with a weighted average of intensity values from nearby pixels. Two pixels can be close to one another, that is, occupy the nearby spatial location, or they can be similar to one another, that is, have nearby values, possibly in a perceptually meaningful fashion. as shown in Figure (3.2) and equation (3.3).

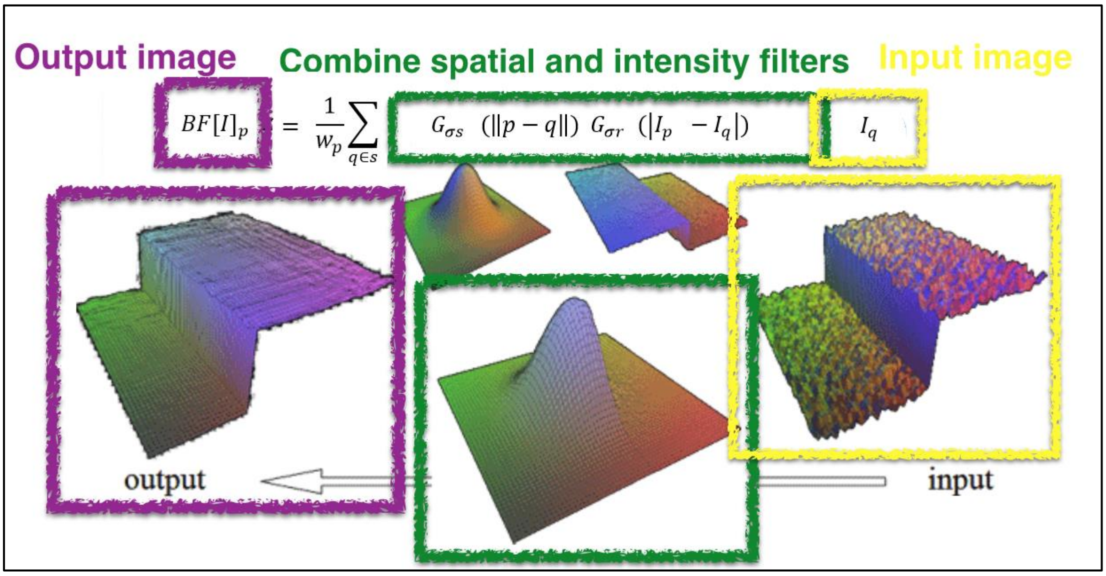



Figure 3.2 Idea underlying bilateral filtering

$$BF[I]_p = \frac{1}{w_p} \sum_{q \in s} G_{\sigma s} \; (\|p-q\|) \; G_{\sigma r} \; (|I_p - I_q|) \; I_q \qquad (3.3)$$

Where:

*Space σs:* the spatial extent of the kernel, size of the considered neighbourhood.

*Range σr:* “minimum” amplitude of an edge.

$w_p$: Normalisation factor

*q:* are the coordinates of the current pixel to be filtered

*s:* is the window centered in q.

$I_q$: is the original input image to be filtered

$BF[I]_p$ : is the filtered image

Figure 3.3 Example of separating seams (red lines)

### 3.3.3 Parameter Selection

The parameters of our algorithm are the number of slices r for the medial seam computation, the smoothing parameter b of the cubic spline filter and the standard deviation $\sigma_s$, and range $\sigma_r$ is “minimum” amplitude of an edge of bilateral filter for the gradient image computation and w bilateral filter window size. In Table (3.1), we show the selected values for the above parameters. All the parameters of our algorithm are assigned the same values that have been used in the method descrined in [77].

| Parameter | value |
|---|---|
| R | 4 |
| $\sigma_s$ | 3 |
| $\sigma_r$ | 0.1 |
| W | 5 |
| B | 0.5 |

Table 3.1 Parameters Selection

### 3.4 Experiments

In this section, we analyse the performance of our algorithm. Our datasets include Arabic, English, and Spanish handwritten document images. We apply our algorithm to the datasets of [77], [78], which is organised in four collections and contains 215 historical manuscripts in different languages. We have evaluated our system using 96 Arabic pages (2043 lines) from Juma Al-Majid Center for Culture and Heritage [89], 70 pages (1229 lines) from Wadod center for manuscripts [90], 40 pages (391 lines) from a 19th-century master thesis collection in the American University of Beirut (AUB) [91] and 9 pages (123 lines) from Thomas Jefferson manuscripts located at the Congress Library alongside with data from University of Qatar. Moreover, the images were selected to have multi-skew, touching/overlapping components and both regular and irregular spacing between lines. The ground truth for the grayscale dataset has been manually generated by defining the region of each text line. Finally, we compare our algorithm with the state-of-the-art method of [77], [78]. The complete list of the dataset is shown in Table (3.2).

| Dataset | Page | lines |
|---|---|---|
| Al-Majid- | 96 | 2043 |
| Wadod | 70 | 1229 |
| AUB | 40 | 391 |
| Thomas Jefferson | 9 | 123 |
| Qatar Dataset | 5 | 93 |

Table 3.2 Details of The Datasets Used In Our Experiments

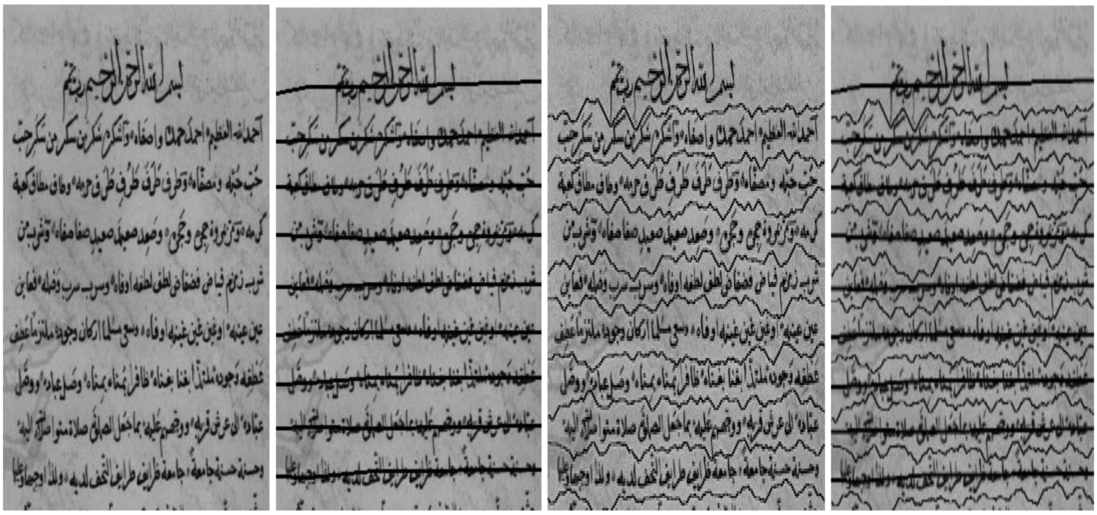
**3.5 Results**

As mentioned above, we have evaluated the proposed system on low-quality historical manuscripts using our datasets: Wadod, University of Qatar, AUB, Al_Majid, and Thomas Jefferson. The first evaluation of the text line extraction experiments is done manually by visually comparing the generated separating seams with the available ground truth. For the purposes of the manual evaluation, we distinguish between two types of seams, according to their accuracy: the first one that passes between two consecutive text lines without cutting through any text components. The second type which cuts through letter components or assigns punctuation marks to the wrong line(s). These seams contain some false information about text line parts, but they are not highly inaccurate. In our algorithms, the obtained performance using the first type (separating seams) 98.5%, and in the second type (medial seems), the amount of error is 1.5%. As we present below, the experiments have shown promising results and the best performance was achieved.

Figures (3.4) to (3.11) show the results obtained by our developed algorithm with the datasets of [77], [78]. For all the figures, label (a) represents the original image, label (b) illustrates line extraction process using the medial seams, label (c) shows the results of line extraction using the separating seams while the last label (d) illustrates the extracted line process by using medial seams and separating seams together.

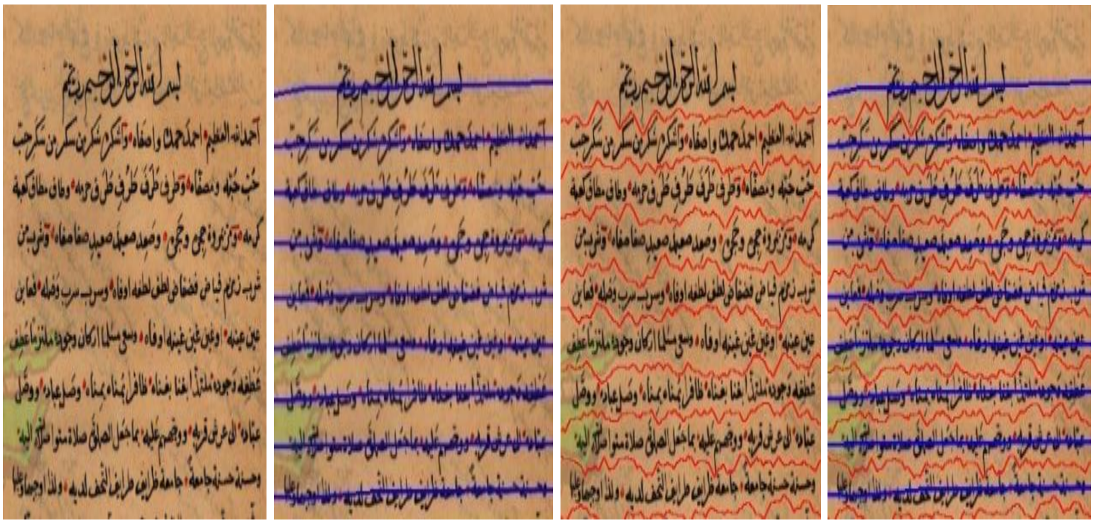
The results below demonstrate that the proposed algorithm enables the separating seams to split touching components along the path passing between the lines and separate them, not necessarily on the optimal position. Moreover, in a dataset such as the Arabic language dataset that includes many dots and diacritics. The place of the dots or diacritics can be closer to the above or below text line. The results suggest that our proposed algorithm is capable to successfully to detect them. As can be seen, the results below show the average performance of our algorithm using various datasets mentioned above of different qualities. All the figures from (3.4) to (3.11) present samples from the tested datasets. It performs well independent of the used script and manages to generate very good results for languages that include delayed strokes, dots, and diacritics. In Figure (3.4) and Figure (3.5), the attractive result has been obtained using Al_Majid dataset grey and colour, the result obtained 99.50% as indicated in Table (3.3). In Figure (3.6) and Figure (3.7), the best results have been achieved using AUB dataset grey and colour, the result obtained 99.80% as indicated in Table (3.3). In contrast, Figures (3.8-b) and (3.8-c) illustrate the results obtained based on the WADUD dataset. The seams cut through the text line and assign word parts to the wrong line. But the overall performance of this dataset (WADUD grey and colour) was 99.30% as indicated in Table (3.3). Figure (3.10-c) illustrates the results of the Thomas Jefferson dataset. The seams that cut through letter component or assign punctuation marks to the wrong line. These seams contain some false information about text line parts, but they are not highly inaccurate. The performance was 98.02%.

Finally, the results obtained based on Qatar university dataset was 97.50%



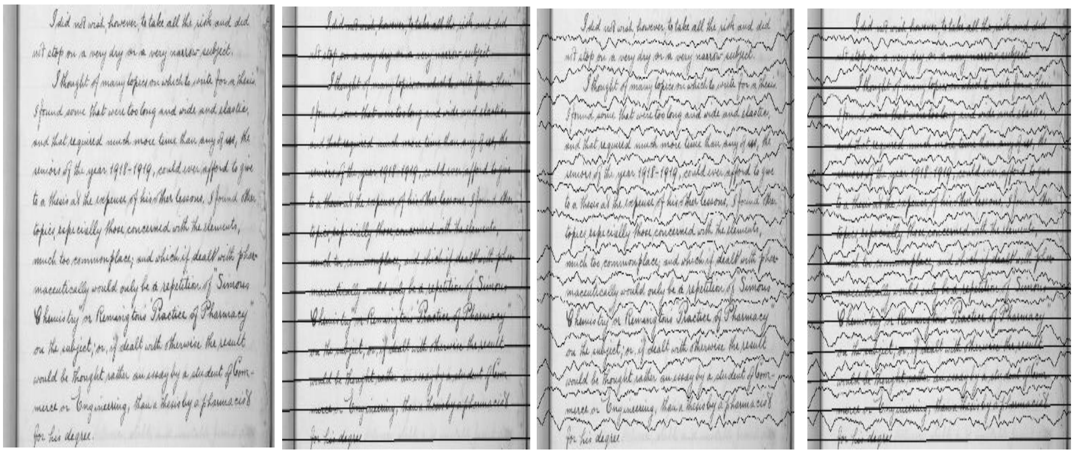
(a) (b) (c) (d)

Figure 3.4 The result of our algorithm: Al_Majid - gray image



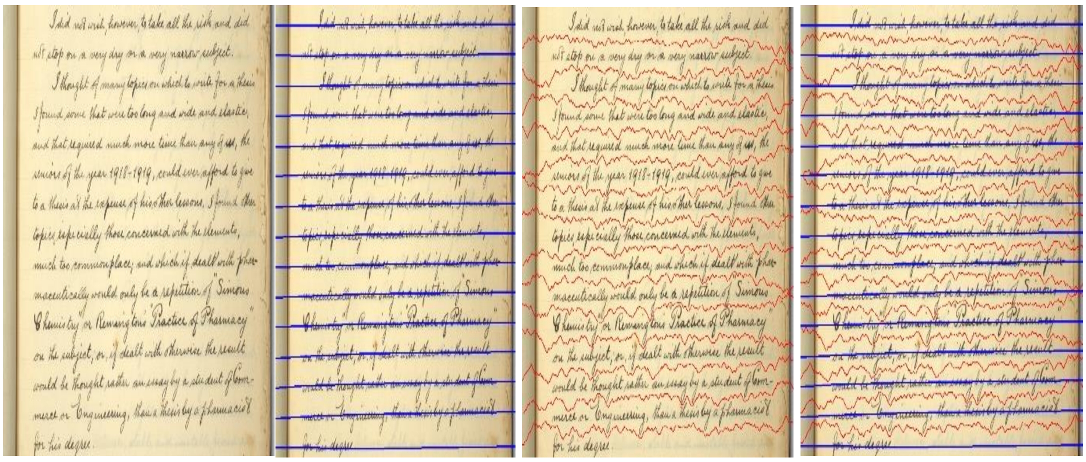
(a) (b) (c) (d)

Figure 3.5 The result of our algorithm: Al_Majid - colour image



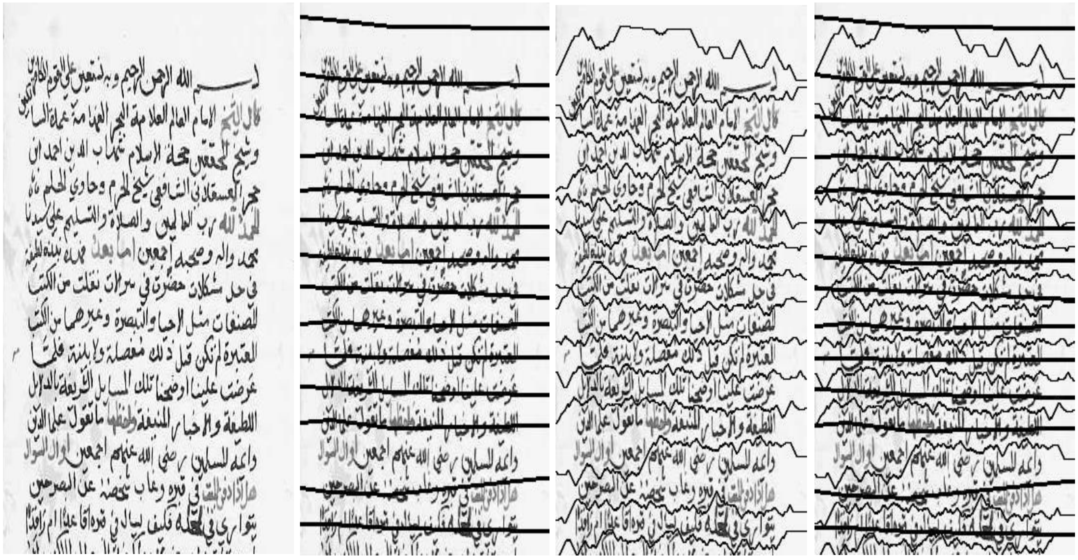
(a) (b) (c) (d)

Figure 3.6 The result of our algorithm: AUB - gray image



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

Figure 3.7 The result of our algorithm: AUB - colour image

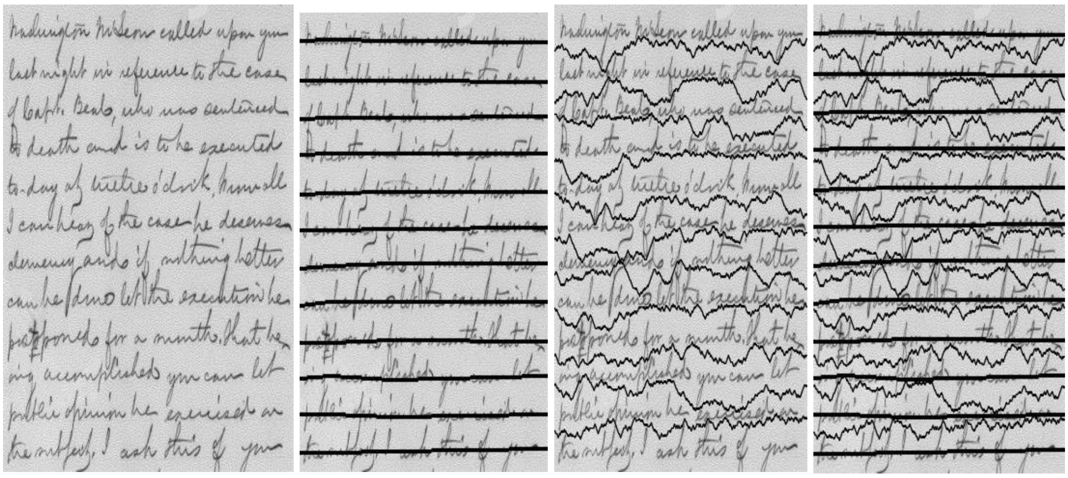


(a) (b) (c) (d)

Figure 3.8 The result of our algorithm: WADOD - gray image

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

Figure 3.9 The result of our algorithm: WADOD – colour image



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

Figure 3.10 The result of our algorithm: Thomas Jefferson image

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

Figure 3.11 The result of our algorithm: Qatar University data

All the results obtained, which are listed in Table (3.3), clearly show the performance of our algorithm compared to two other algorithms that have been developed before in [77] and [78]. As mentioned above, the results of our algorithm show similar peak performances, but the most notable result is the performance using AUB dataset that slightly outperformed the other results. Figure (3.12) shows the comparison of our algorithm performance with the algorithms in [77] and [78].

| Dataset | Performance (%) of Approach developed by | | |
|---|---|---|---|
| | **Ours** | **Authors** [77] | **Authors** [78] |
| Al-Majid- | 99.50 | 99.30 | 97.56 |
| Wadod | 99.30 | 99.04 | 98.35 |
| AUB | 99.80 | 99.75 | 98.05 |
| Thomas Jefferson | 98.02 | 97.75 | 98.55 |
| Qater Dataset | 97.50 | - | - |

Table 3.3 Comparison with the approaches in [77] ,[78]

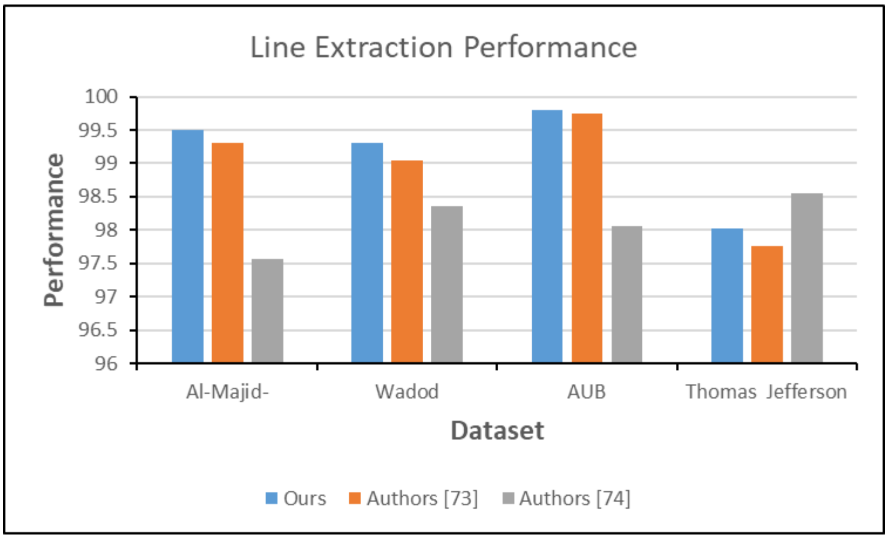



Figure 3.12 Comparison with the approaches in [73] ,[74]

### 3.6 Summary

This chapter proposed a novel algorithm for the segmentation and detection of text lines of grey-scale and colour historical documents. A text line extraction algorithm is developed using bilateral filtering in order to compute separating seams. The use of Gaussian filtering in uniform areas of an does affect object borders. In addition, the use of a bilateral filter results in in a much enhanced image. The proposed algorithm has been implemented to deal with both grey-scale and colour historical documents. The performance of our algorithm is dependent on the medial seam computation and separating seam computation using bilateral filtering. The experimental validation of the algorithm has been carried out using different datasets and the results obtained suggest that our proposed technique yields attractive results when compared against a few similar algorithms.

The next chapter proposes to deploy a combination of both approaches using Oriented Basic Image features and the concept of graphemes codebook.

# Chapter 4

# Writer Identification using the Concept of Bag of Words

## 4.1 Introduction

There are two approaches of writer identification systems: statistical and model-based approaches. A statistical approach analyses statically the set of extracted features from the handwritten text [92], on the other hand, the model-based approach uses graphemes which are limited samples of handwritten strokes [24], [93]. Both approaches consist of two phases: a feature-extraction phase and a classification phase. In the feature extraction phase, the features are extracted from the handwriting scripts and the generated features are then analysed for their distinctive power before stored as a single feature vector. In the classification phase, the resulting feature vector is matched and assigned to different classes that best represent the authors.

The main objectives of this chapter are twofold: (i) to develop an effective feature extraction that best distinguishes handwriting patterns. To achieve this, a combination of a multiscale feature extraction with the concept of grapheme is judiciously carried out to capture and extract the several discriminating features such as handwriting curvature, direction, wrinkliness, and various edge-based features and (ii) develop a novel text-independent writer identification system for offline Arabic and English writings captured as scanned images.

The remainder of this chapter is organised as follows: Section 4.2 discusses how we can improve the approaches to identify writers. The proposed approach is then detailed in Section 4.3. Section 4. 4 discusses the experiments carried out to obtain results and their validation including a comparative study against some previous works. Finally, Section 4.5 concludes the chapter.

### 4.2. Approaches to handwriter Identification

Several statistical and model based features were proposed in [94] where the authors proposed an approach to improve the statistical feature extraction using an edge hinge distribution. Moreover, the authors explored a combination of this feature extraction approach with a codebook of the graphemes. The system was evaluated using the Firemaker database, which consists of 250 writers with 4 pages per writer. The work [95] used a writer identification system using the concept of oriented Basic Image Feature Columns (oBIF Columns) and the authors proposed how a texture-based scheme can be enhanced by encoding a writer's style as the deviation from the mean encoding for a population of writers. In their work described in [96] the authors have proposed a system using a texture based approach for the identification of writers from offline handwritten images. The proposed method was implemented by dividing a handwriting script into small fragments where each fragment was processed individually as a texture. The authors used both Arabic and English text from IFN/ENIT and IAM databases to evaluate the performances. [97] proposed an approach for writer retrieval and writer identification based on texture features. In both cases, a codebook was generated using the Scale Invariant Feature Transform (SIFT) from different pages of the handwriting. Then, a histogram was generated and used to identify a writer or retrieve the documents of one particular writer. The IAM dataset was used for the evaluation resulting in an identification rate of 90.8%. Tang and Wu (2013) proposed two feature extraction methods: the stroke fragment histogram (SFH) based on a codebook and a local contour pattern histogram (LCPH) generated by tracking the points on the contours of the handwriting images. Identification rates of 91.3% and 85.4% were obtained for SFH and LCPH, respectively. Another approach to evaluate the identification performance of five highly discriminating features was proposed in [98]. The five classes of features investigated are slant and slant energy, skew, pixel distribution, curvature, and entropy. The work [99] presented a writer identification system using graphometrical and forensic features using an Artificial Neural Network (ANN) for the classification task. A database of 100 users with 10 samples per subject was constructed and the system achieved an identification performance of 94.6%.

The authos in [100] proposed a text-independent writer identification system using the histogram of the codebook shapes to generate a feature vector for each manuscript. Furthermore, the technique uses cursive handwritings with rich content of dissimilar shapes present in the handwriting connected components. Only part of the connected components was used to avoid complex patterns. Two approaches were used to extract codes from the contours. First, using the actual pixel coordinates of contour fragments. Second, using a linear piece-wise approximation using the lengths and angles of the segment. The two methods were evaluated using two English and three Farsi handwriting databases. The authors concluded that both methods have shown promising results. However, the performance of the later method is better than the first method.

Furthermore, a writer identification system for offline text-independent Arabic language was proposed in [101]. The main idea of this method uses a beta-elliptic model in order to generate a synthetic codebook. In this algorithm, a feature selection was proposed to reduce the codebook's size where the feature extraction is performed using a template matching approach. The authors in [4] proposed a handwriting-based identification system for Arabic handwritten documents. Their proposed method consists of two main stages: first, the system processes each handwritten image and extracts features such as edge-direction, edge-hinge, and run length features. Then, these features are fed to a Multiclass SVM (Support Vector Machine) for classification. To validate the approach, the method was trained and tested on a large database of Arabic handwritings written by 1000 different writers. The authors reported that the best result was achieved when combining run-length and edge-hinge features achieving a classification rate of 84.10%.

Finally, the authors in [69] proposed a writer identification method using a codebook extension model with an ensemble of codebooks in which a kernel discriminant analysis using spectral regression (SR-KDA) was deployed as a dimensionality reduction technique to avoid the over-fitting problem. Two datasets were used in evaluation with a single codebook and using multiple codebook sizes. Furthermore, the authors conclude that the proposed method is competent when compared against existing methods.

As discussed previously most existing handwriter identification systems either use a statistical extraction method or a model-based approaches with much efforts made to the feature selection and dimensionality reduction using robust classifiers. It is also known that both feature extraction approaches have useful advantages including some limitations on their own. This chapter proposes to combine the two approaches and develop a novel statistical and model-based feature extraction approach in order to improve the recognition performance.

## 4.3. Proposed Methodology

As mentioned previously, most existing handwriting identification systems are based on two approaches: a statistical approach or a model-based approach. These approaches have some limitations that lower the performance of the handwriting identification system. Therefore, this chapter proposes an automatic handwriting identification system by combining both approaches. This is achieved by fusing Oriented Basic Image (OBI) features with a codebook of graphemes in order to improve the recognition performances of the work described in [69]. The IAM (English) and ICFHR-2012 (Arabic) databases have been used for evaluation especially that both have several discriminating features such as handwriting curvature, direction, wrinkliness and some edge based features which require an efficient feature extraction strategy. In this chapter and due to the uniqueness of the features, we have investigated various methods including SIFT, Speed Up Robust Features (SURF) and OBIs. An initial investigation has shown that OBI method outperforms others, therefore this method has been used as a base. We extract OBI features using a multi-scale approach with local symmetry and orientation. We use different orders and directions of multi-scale Gaussian derivative filters to generate a number of features. Then, orientations and scales of the features a histogram is generated based on the symmetry, which, once normalised, generates the final feature vector. The generated feature vector is then combined with a grapheme based codebook to investigate the system identification performances.

Furthermore, to maintain the system resources and increase the system operation speed, one needs to decrease the length of feature vectors. The Kernel Principal Component Analysis (KPCA), reduction technique, has been used because of its simplicity and effectiveness compared to other methods. For classification, various classifiers were initially tested including K-Neighrest Neighbour, Support Vector Machine (SVM), and Neural Networks. The 1-NN with Euclidian distance has been adopted as it provides the most effective results. The experimentation has been carried out using the two datasets (English and Arabic) using a single codebook first. However, the results obtained depicted below have shown that the performance was marginally similar to that proposed in [69]. Therefore, to further improve system performance, a multiple codebook approach has been investigated. In this chapter, we propose a writer identification system based on combining different OBI features with different graphemes codebooks. The system overall stages are illustrated in Figure (4.1) and will be discussed in the next sections.

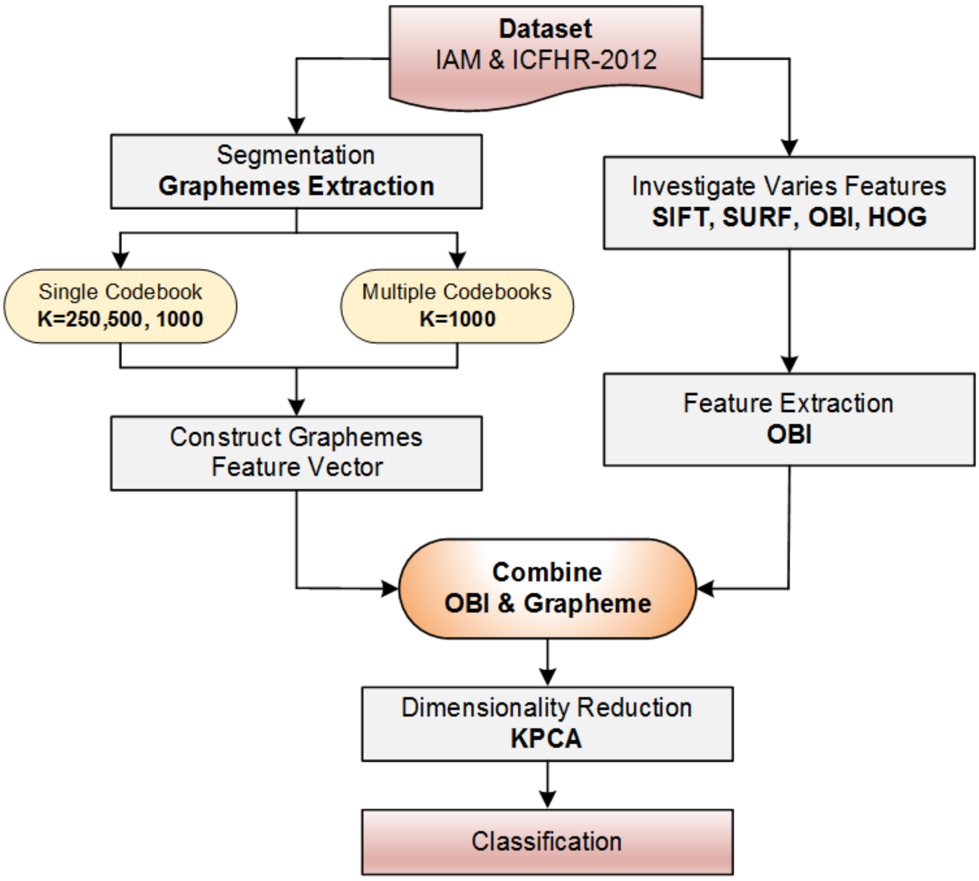



Figure 4.1 The system overall diagram flow.

### 4.3.1. Datasets

To evaluate the performances of the proposed approach experiments were carried out using two datasets: the IAM [68] English dataset and the ICFHR-2012 [71] Arabic dataset.

### 4.3.2 OBI Feature Extraction

Basic Image Features (BIFs) are an efficient method used for encoding images. The main idea of this algorithm is based on the responses of a bank of six DtG filters that have been used to classify each pixel of an image into one of seven BIFs and producing a primal sketch of the input image. This step is mainly depended on local structures and symmetries. As shown in Figure (4.2), the calculation process of BIFs is treated and examined based of the value of a scale parameter called the standard deviation of the DtG filters ($\sigma$) and a threshold parameter ($\varepsilon$) dictating the part of an image that should be considered as 'flat' [102] .

Moreover, the BIFs algorithm is also used to encode compact representations of images such as 7-bin histograms. In this case, the seven classes will be increased to 23 by using the quantisation of the non-rotationally symmetric features [102].

In this work, various feature extraction methods have been investigated as discussed previously and the findings have shown that the OBI method outperforms others thus method has been adopted in this chapter. Figure (4.1) shown the structure of our method. We start by describing the OBI method since it is an important component of the algorithm. BIFs consist of texture-based patterns encoded as images as follows. In an image, each location is classified into one of seven types using an approximate local symmetry type as described in [95] and [103]. The local symmetry types are a flat, dark and light rotational, dark and light line, slop and saddle-like patterns. To classify the patterns, a bank of six derivative-of-Gaussian filters is used: one $0^{th}$ order filter, two $1^{st}$ order filters, and three $2^{nd}$ order filters. To compute the BIFs two tuneable parameters are necessary: a filter scale parameter σ and a threshold value $\varepsilon$ These will effectively classify the locality as flat, or as one of the other symmetry types. For example, by increasing the value of $\varepsilon$ to a high value, the image will be classified as flat [104].

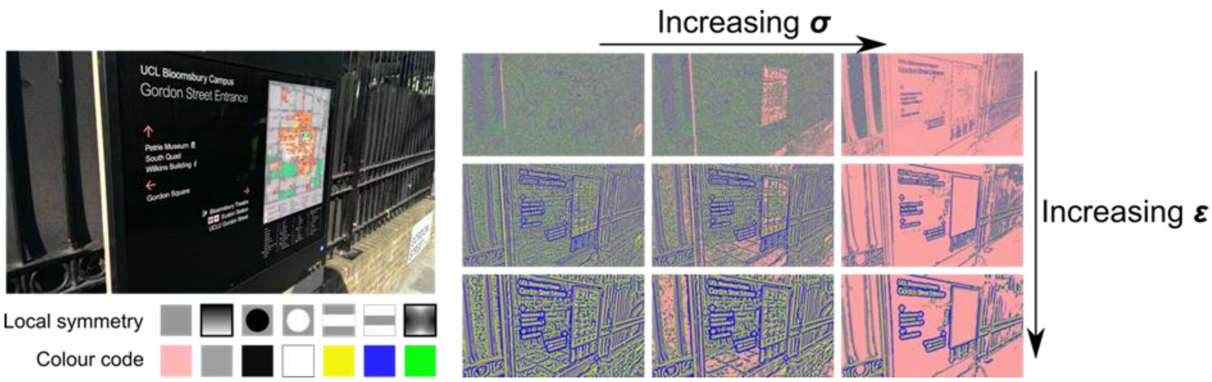



Figure 4.2 oBIFs computation for scale parameter σ and $\varepsilon$ [95]

Modification and extension to the BIF algorithm have been proposed through a combination of local orientation with local symmetry type resulting in the generation of oriented BIFSs (oBIFs). In this case, the value of n, which represents the orientation quantisation level, will enable the extraction of the possible orientations values depending on the local symmetry type [95]. For example, by setting the location to dark or light or a flat type there will be no orientation exhibited. On the other hand, by setting the location to a dark line, light line or saddle-like types will lead to the specification of n possible orientations. Finally, a slope type location will specify 2n orientations since a slope has also a further directional feature. Consequently, this results in a set of (5n+3) features [104]. The oBIF calculation is described in Figure (4.3).

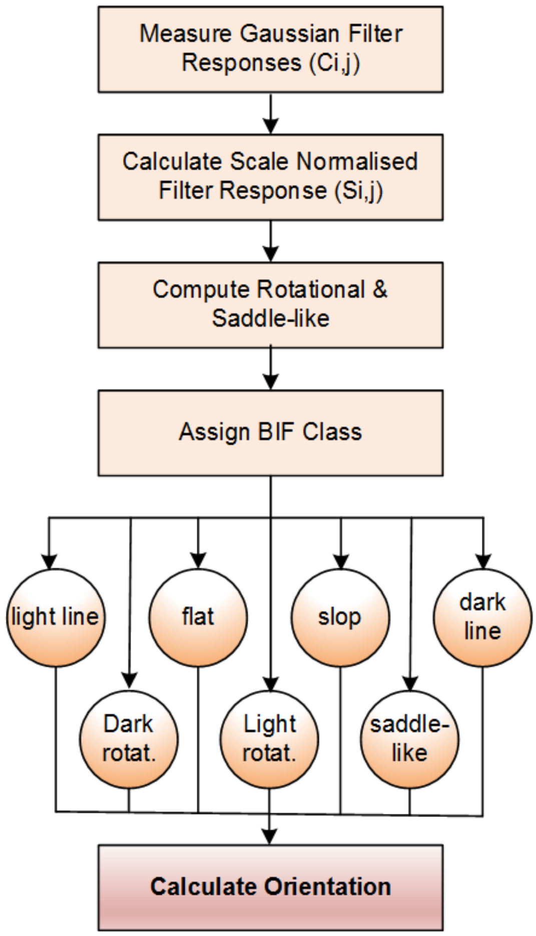



Figure 4.3 Method to calculate oBIF

### 4.3.3 Codebook Extraction

The grapheme codebook method has been shown to be a useful technique in various pattern recognition problems including writer identification. It works by first extracting the graphemes, which can be defined as small pieces or segments of a character. One simple and effective method to extract the graphemes can be done by splitting the connected components of the written text. This can be carried out by using a suitable algorithm such as the ink-trace minima heuristic method (the ink is cut at the minima in the upper contour for which the distance to the lower contour is comparable to the ink-trace)  shown in Figure (4.4) [36].

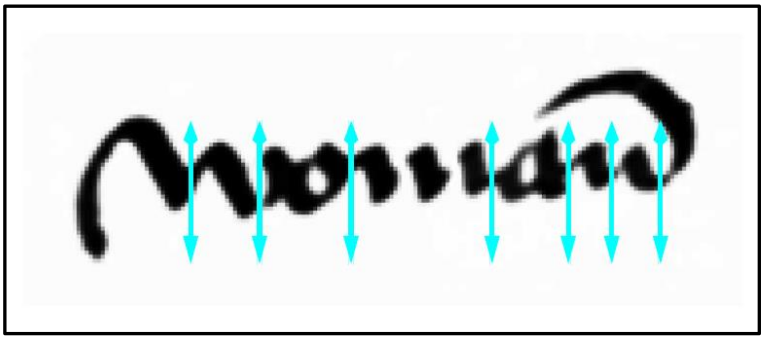


Figure 4.4 Grapheme splitting points by the minima heuristic [24]

### 4.3.4 Grapheme Codebook Approach

In the literature, there are a wide variety of approaches that have been investigated and used in writer identification task. One of these approaches is the grapheme codebook method [24], that is chosen and used in this part of the research as an important phase to analysis datasets in order to achieve improved identification performance.

This method depends on the text segmentation into graphemes, which are small pieces of character-scale text, have been created by partitioning a cursive ink trace heuristically. Usually, the main purpose of this method is to provide character-like segmentation of the text, but we didn't need to use the meaning of the character or the grapheme, which means all types and the shapes of the characters (whole, partial or merged) will be equally valid. Moreover, a good heuristic will contribute to providing robust segmentation, for instance, trying to give an identical ink trace, this will create similar output graphemes. So, this will let us compare of distribution of ink trace forms that are created by a writer [24].

The main pre-processing steps that are required for the codebook technique are illustrated in Figure (4.5). In the first step, we need to apply binarising tasks over the input image. After that, all the connected-components must be extracted, which means, we need to look for all connected parts of ink trace pixels. These components are split up into graphemes and then must be adjusted to a specific size in order to make a comparison process. In this case, all images that are used in the training and test groups can be represented by their constituent graphemes, and then the entire dataset is represented by the union of the constructed graphemes. Therefore, from these grapheme collections, a typical number between 100-1000 has been chosen in order to produce a global codebook. It is the reference basis that will be used to measure and check the images in the dataset.

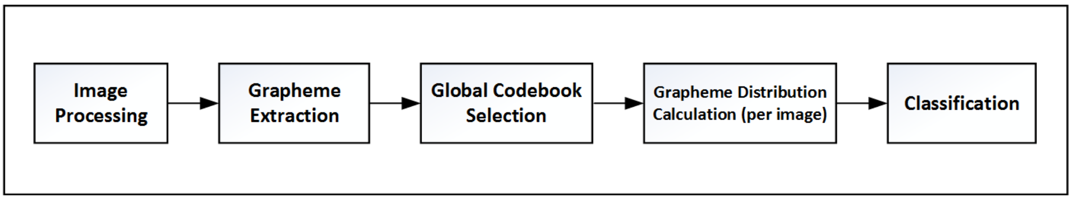



Figure 4.5 The summary of the grapheme codebook method

After that, the features are measured and then we can compute the feature vector for all the images in the dataset. For any testing image, the grapheme is created and then have to compare with all the graphemes in the codebook, and then should be tallied against one of the images that match nearly. When this step is achieved for all graphemes in an image, the tally should be normalised to sum to 1, which will yield a probability distribution. All the previous steps are to create the feature vector of an image with respect to one of the codebooks that have been selected. At the end of this step, all the image feature vectors have been determined and there is another optional step that is feature selection or extraction that may take place. At the end of this process, the last step is classification, which is aimed at comparing the features vectors that belong to the training set with other features vectors that are calculated from unknown images in order to define their most likely writer.

The generated graphemes of an image would appear as an unordered bag of patterns and will be used to extract the codebook which will act as a reference set of graphemes. This descriptor will be used to determine a 'shape alphabet' with which to describe each image. There exist a number of codebook generation methods in the literature based on various criteria [100], [101], [105]. Various codebook selection methods can be used to extract codebooks depending on the application at hand.

In this chapter, we proposed to generate a codebook of the dataset to efficiently represent the data being tested so that the shapes to be recognised are closely tuned to the shapes used by the authors of the scripts. To achieve this, a selection approach to collect the graphemes by a shape-based similarity approach using a Kohonen Self-Organising Feature Map (SOFM) proposed in [106] has been used by specifying the number of clusters to be related to the size of the generated codebook. Furthermore, we propose to use the cluster centres for the codebook where each one is chosen as a representative of its cluster of similar graphemes. Extensive training is required by the SOFM in order to ensure convergence to a layout so as to generate the most effective codebook. Once the creation of the codebook is done, feature extraction is then required.

As mentioned above, the proposed approach presented in this chapter can be considered as a further development of the previous work of [69]. For a fair comparative study, the codebook generation steps of their work were followed. Therefore, one of the essential steps is to measure and investigate the effect of combining multiple codebook features on the identification performance rates. The representation of multiple codebooks can be defined as:

$$\mathrm{Y} = \sum_{j=1}^{n} P_j \qquad (4.1)$$

where:

$\mathbf{Y}$: the number of graphemes extracted from the whole training set.

$\boldsymbol{n}$: the number of partitions of the graphemes.

$\boldsymbol{P_j}$: one partition of the graphemes.

From equation (4.1), the grapheme features are divided into $\boldsymbol{n}$ partitions $\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3} \ldots \ldots \boldsymbol{P_n}$ each of size $\boldsymbol{W}$. A popular tool of creating a codebook can be achieved by using k-means clustering, which is an unsupervised learning algorithm [69], [100]. Therefore, the features of each grapheme partitions $\boldsymbol{P_j}$ have been clustered using k-means clustering algorithm. Once done, one needs to find the centres $\boldsymbol{C_k}$ of each cluster so that each writer sample of the training dataset can be assigned into a codebook (cluster) of size $\boldsymbol{Q_k}$, which can be done by mapping the grapheme features of the sample image to the nearest centre $\boldsymbol{C_k}$. In this work, at the first stage, the proposed approach investigates the identification performances based on sa ingle codebook grapheme. In the second stage, the effect of combining multiple codebook grapheme features is investigated and the findings discussed.

### 4.3.5 Combining OBI and Grapheme Features

The main idea of the proposed identification system relates to the fusion of OBIs and Grapheme features resulting in a large feature vector. Therefore, a reduction of the resulting high dimensionality vector is crucial in order to select those features with high discriminative power while at the same time speeding up the recognition process. The following presents the proposed approach used to address this issue.

### 4.3.6 Dimensionality Reduction

Dimensionality reduction is a process to extract the most discriminative information from a high-dimensional data set. The concept is to compact the raw data into a more condensed form so as to reduce both the high dimensionality of the feature vector and the computational complexity while still keeping intact the accuracy of the recognition [107]. However, the performance will be affected if noisy or faulty input data are considered since the removal of redundant data should not degrade the performances. This process can be divided into feature selection and feature extraction. In the first approach, one attempts to detect a subset of features of the data using three strategies: filter, wrapper, and embedded methods. The main idea of the second approach is to reduce the high dimensional data into a space of fewer dimensions and both linear or nonlinear techniques can be deployed.

Linear dimensionality reduction techniques are useful in many pattern recognition problems as a tool to support the analysis of high dimensional datasets [108]. It noted that linear methods may not be appropriate for use directly for the identification of handwriter since data is non-linear. However, they are simple and can be modified and tuned for nonlinear problems. In this chapter, we have adopted such an approach as described in the following. Linear dimensionality reduction methods work by generating a low-dimensional linear data of the original high-dimensional data while maintaining the most discriminative features of the data. Principal Component Analysis (PCA), which is a very popular linear technique for dimensionality reduction, implements a linear mapping of the data to a lower-dimensional space ensuring that variance of the new data in the low-dimensional space is maximised [107], [108]. PCA can be used in a nonlinear approach through the kernel trick. The output method can be employed to construct nonlinear mappings that maximise the variance of the data.

The resulting approach, termed kernel PCA (KPCA), operates in a similar fashion as in conventional approach with the main difference being the use of a nonlinear mapping which maps each given data point onto an abstract function. In other words, KPCA technique implements PCA with some extra functionality of the kernel trick [107], [109]. At the first step, PCA starts by calculating the covariance matrix of the image matrix as shown by equation (4.2).

$$C = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x_i}\ \mathbf{x_i^T} \qquad (4.2)$$

Where:

$\mathbf{x}_i \in R^d$ . $d$-dimensional random vector expressed as column vector.

It operates by diagonalising the covariance matrix $\mathbf{x_i^T}$ .

KPCA starts by calculating the covariance matrix of the data after being converted into a higher-dimensional space as indicated in equation (4.3).

$$C = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\Phi}(\mathbf{x_i})\ \ \boldsymbol{\Phi}(\mathbf{x_i})^{\mathbf{T}} \qquad (4.3)$$

Where:

$\boldsymbol{\Phi}(\mathbf{x_i})$ is embedding of data items $x_i$. Let $\boldsymbol{F}$ be a feature space induced by nonlinear mapping $\boldsymbol{\Phi}$: $R^d \rightarrow \boldsymbol{F}$.

It then maps the converted data from the previous step onto the first **k** eigenvectors. It utilises the kernel trick to factor away much of the computation, such that the whole procedure can be executed without evaluating Φ(x). Obviously, Φ must be selected such that it has a known corresponding kernel.

### 4.3.7 Classification

The classification, which is the final stage of the system, uses K-nearest neighbour (k-NN) classifier. This classifier has been utilised in many pattern classification problems and is very useful for measuring the distances between the test data and each of the training data in order to determine the final classification result [29]. Moreover, this algorithm is a simple yet effective classifier because it can use different distance measures such as Euclidean distance, Chi-square distances, and Manhattan distance.

In this work, we have investigated these methods and the results obtained are as follows: Manhattan distance = 80%, Euclidian distance = 96.05%, and Chi-square distance = 73%. This has allowed us to adopt the Euclidian distance in our work since it outperforms the other metrics. For experiments in the statistical approach, we have used 650 writers from the IAM dataset. In our combined model, matching is carried out using equation (4.4) as follows:

$$\boldsymbol{E_i} = \sqrt{\sum_{j=1}^{k}(\boldsymbol{M_{ij}} - \boldsymbol{V_i})^2} \qquad (4.4)$$

where:

$E_i$: the final distance between the input sample and model$i$

$k$: is the number of features in the features vector

$M_{ij}$: is the $j$th feature of model $i$,

$V_j$: is the $j$th feature of the input sample feature vector.

## 4.4 Experimental Results

To evaluate the performances of the proposed approach, a set of experiments has been conducted. As mentioned above, the experimentation has been carried out using the

IAM dataset for English handwriting and ICFHR-2012 dataset for Arabic handwriting. The codebook is generated and used in the system for both English and Arabic datasets. The evaluation was carried out first using a single codebook followed by a multiple codebook method. The results obtained were compared against the results reported in a similar approach [69].

### 4.4.1 Single Codebook

In this experiment, KPCA has been used as a nonlinear dimensionality reduction technique to produce low dimensional data in order to overcome the over-fitting problem and save the system resources while speeding up the execution time. This experiment was conducted on the IAM English dataset. To provide similar evaluation conditions in [69], their experimental steps have been followed for the sake of fair analysis. Initially, 188k graphemes were generated from the handwriting training set of 127 writers. Then, a codebook of size 250 using k-means clustering (k = 250) was created. For each input sample from the testing set, the writer descriptor for that sample was extracted from the generated codebook. The handwritten target sample is then compared against 1313 other samples. The classification process using a 1-nearest neighbour classifier (using Euclidian distance) is used to evaluate the writer identification performance. Table (4.1) depicts the performance results obtained under different codebook sizes ranging from 250 up to 1000.

As shown in Table 1, with a codebook size of (250-500- 1000) using *k*-means clustering (*k* = 250, 500 and 1000), one can notice that the best result with a recognition rate of 88.01%, which is achieved when the codebook size is 1000. On the other hand, this result is better than the result of 81% with a codebook of 1000 as in [69]. The results obtained are shown in Figure (4.6) and clearly demonstrate that our results are attractive in the case of a single codebook.

| Codebook size | Top-1 (%) | | Top-5 (%) | | Top-10 (%) | |
|---|---|---|---|---|---|---|
| | Our work | previous work | Our work | previous work | Our work | previous work |
| **250** | 87.56 | 80.00 | 90.13 | 88.00 | 93.34 | 93.00 |
| **500** | 87.96 | 80.00 | 91.34 | 89.00 | 94.28 | 94.00 |
| **1000** | 88.01 | 81.00 | 94.16 | 89.00 | 96.45 | 94.00 |

Table 4.1 Comparison of system's performance with previous work [69]

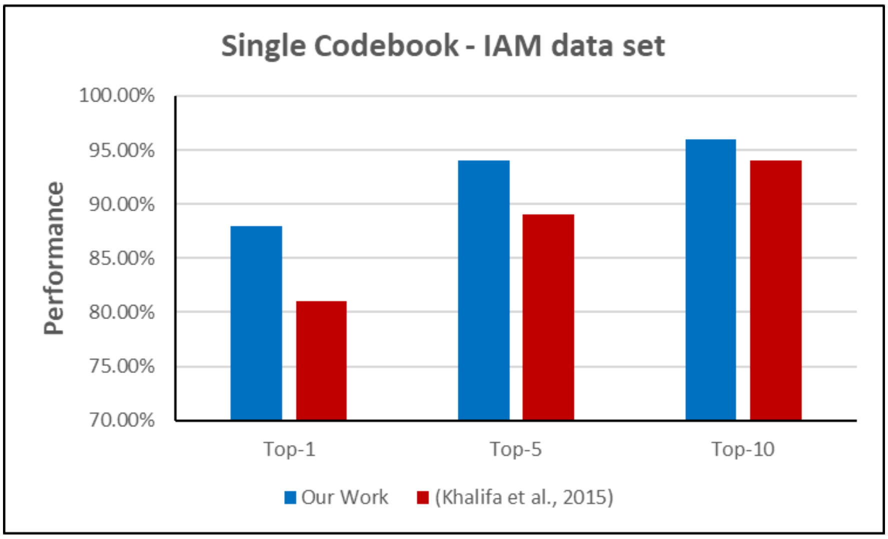



Figure 4.6 System performance for codebook size=1000

### 4.4.2 Multiple Codebooks

In the second part of the experiment, we have investigated the effect of combining multiple codebook features on the identification performance rate of the system. In the first step, the total graphemes size extracted from the whole training data set (English and Arabic) is split randomly into $n$ partitions (as shown above by equation (4.1) without an overlapping of the graphemes (where n=3,4,6,8,10,12,14,16).

Therefore, $\boldsymbol{Y}$ multiple reference codebooks are generated from $\boldsymbol{P}$ for use to investigate and determine the effect of $\boldsymbol{Y}$ codebooks on the system performance. The total size of the graphemes has been randomly partitioned into $\boldsymbol{n}$ parts. Likewise, the experiments have used both English and Arabic datasets and the results will be illustrated and discussed in the next sections.

- **Evaluation of the Performance using English Dataset**

The system performance for Top-1 identification is assessed and compared against the results reported in [69] including the computational complexity. Table (4.2) depicts the results obtained when n is varied. As shown in Table (4.2), the average execution time of our proposed system is around 1.48 sec compared to 31.5 sec in the work of [69] On the other hand, the maximum value of the system performance in their work was 92% compared to 90.41% in our work. Moreover, if we consider the execution time, one can conclude that the use of KPCA is a more effective reduction technique in terms of the execution time factor. This experiment proved that the identification system's performance, as well as the execution time, have been improved compared to the work of [69].

Figure (4.7) illustrates the execution time of our system based on KPCA technique versus the KDA technique given in [69]. From the results obtained, one can notice that the KPCA technique is capable to improve the execution time of the system compared to KDA. Figure (4.8) shows a comparison of the performance for Top-1 between our system and the work in [69].

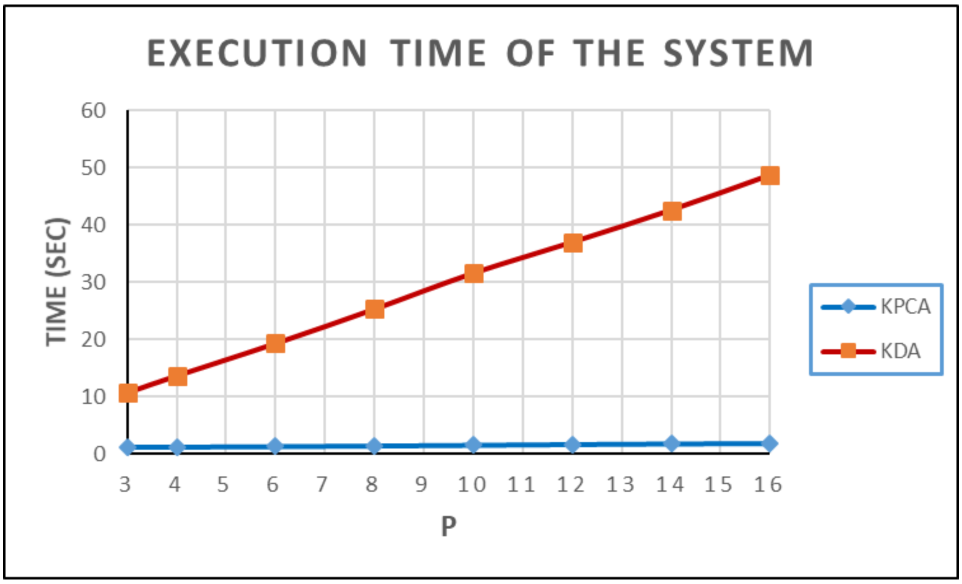



Figure 4.7 Execution time for KPCA vs KDA (English Dataset)

| n | Top-1 Our work | | Top-1 Previous work [69] | |
|---|---|---|---|---|
| | **Performance (%) (Based on KPCA)** | **Execution Time (sec)** | **Performance (%) (Based on KDA)** | **Execution Time (sec)** |
| 3 | 86.34 | 1.13 | 84.00 | 10.64 |
| 4 | 87.46 | 1.16 | 85.00 | 13.59 |
| 6 | 89.34 | 1.25 | 87.00 | 19.24 |
| 8 | 89.95 | 1.35 | 89.00 | 25.23 |
| 10 | 92.03 | 1.48 | 90.00 | 31.50 |
| 12 | 89.57 | 1.60 | 92.00 | 36.90 |
| 14 | 90.25 | 1.72 | 90.00 | 42.50 |
| 16 | 90.41 | 1.79 | 88.00 | 48.60 |

Table 4.2 System performance (Top-1) versus execution time

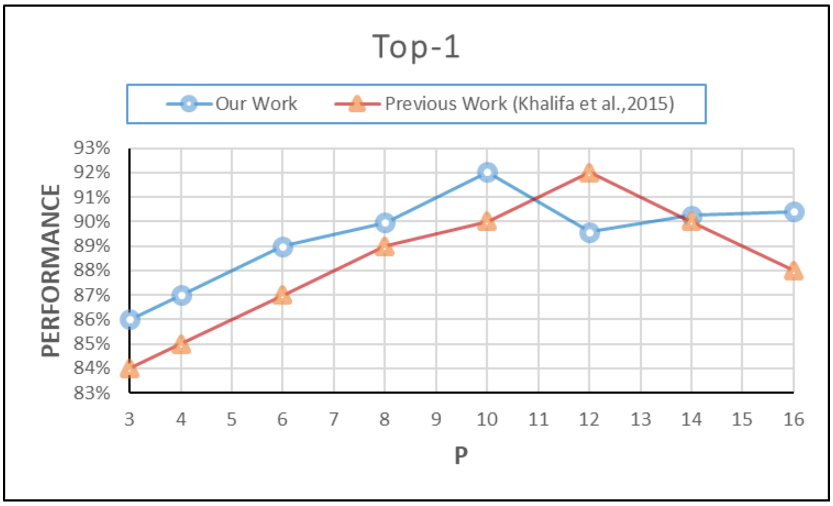



Figure 4.8 Comparison of the performance for Top-1 identification

The results depicted in Table (4.3) present the performance of our developed system for Top-5 and Top-10 identification. In the case of Top-5 identification, it can be observed from the table that when n= 10 and 12, a performance of 95.28% is obtained demonstrating that the proposed technique outperforms the method proposed by [69]. In addition, in the case of Top-10, the performance at n=12 reaches 97.34% compared against 97% in [69]. Although the results are marginally better, and the speed performance is significantly better as shown in Figure (4.9) and Figure (4. 10).

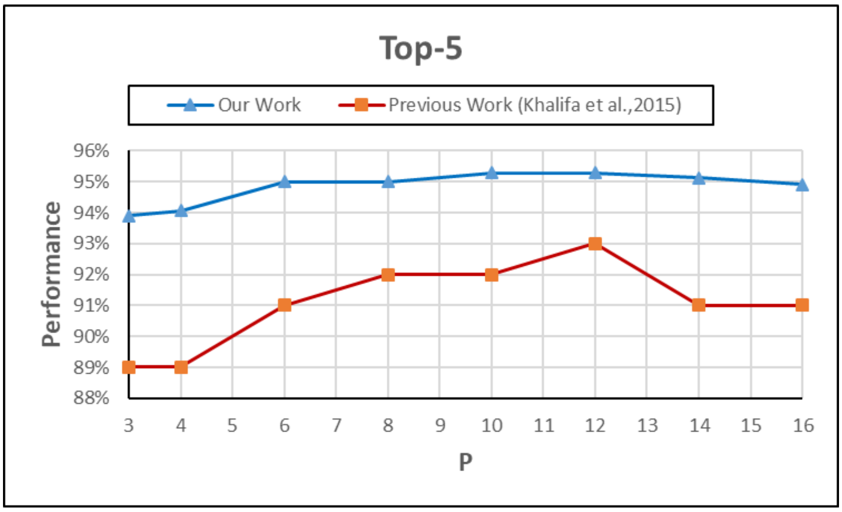



Figure 4.9 Comparison of the performance for Top-5 (English dataset).

| n | Top-5 (%) | | Top-10 (%) | |
|---|---|---|---|---|
| | Our work (Based on KPCA) | previous work* (Based on KDA) | Our work (Based on KPCA) | previous work* (Based on KDA) |
| **3** | 93.90 | 89.00 | 94.90 | 92.00 |
| **4** | 94.06 | 89.00 | 95.43 | 93.00 |
| **6** | 95.19 | 91.00 | 96.42 | 94.00 |
| **8** | 95.48 | 92.00 | 96.67 | 95.00 |
| **10** | 95.28 | 92.00 | 96.57 | 96.00 |
| **12** | 95.28 | 93.00 | 97.34 | 97.00 |
| **14** | 95.13 | 91.00 | 96.27 | 95.00 |
| **16** | 94.90 | 91.00 | 95.89 | 94.00 |

Table 4.3 System performance for Top-5 and Top-10 identification versus to the performance in [69]

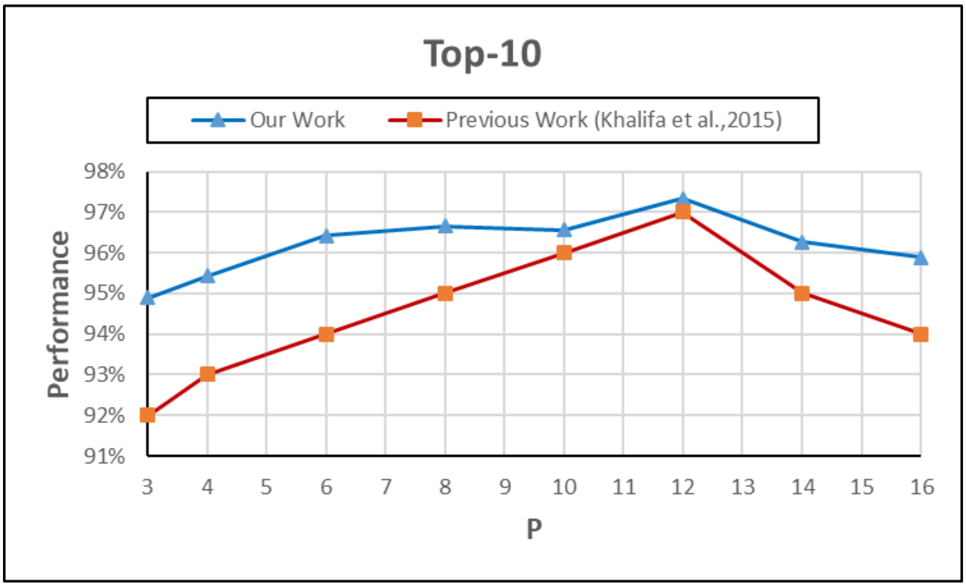



Figure 4.10 Comparison of the performance for Top-10 (English dataset).

➢ **Evaluation of the performance using Arabic Dataset**

In this part of the experiment, the system performance is evaluated and compared against the results given in [69] using the Arabic dataset. The results obtained indicate that the proposed method demonstrates an improvement in the identification rates when compared against the results of [81]. Table (4.4) illustrates the identification results for Top-1 and showing that the maximum identification performance is 96.49% against 95.00%.

Figure (4.11) illustrates the execution time of our system using the KPCA technique versus the KDA technique used in [69]using the Arabic dataset. From the figure, it can be observed that the KPCA technique has improved the execution time of the system compared to KDA. Figure (4.12) shows a comparative analysis of the identification performance for Top-1 of the two methods.

| n | **Top-1** Our work | | **Top-1** previous work | |
|---|---|---|---|---|
| | **Performance (%) (Based on KPCA)** | **Execution Time (sec)** | **Performance (%) (Based on KDA)** | **Execution Time (sec)** |
| **3** | 96.49 | 1.15 | 88.00 | 11.64 |
| **4** | 96.49 | 1.18 | 89.00 | 15.59 |
| **6** | 96.49 | 1.29 | 90.00 | 19.24 |
| **8** | 96.49 | 1.37 | 92.00 | 26.23 |
| **10** | 96.05 | 1.49 | 93.00 | 33.50 |
| **12** | 96.05 | 1.62 | 95.00 | 38.90 |
| **14** | 96.05 | 1.77 | 92.00 | 45.50 |
| **16** | 96.05 | 1.80 | 91.00 | 49.60 |

Table 4.4 System performance (Top-1) versus execution time for Arabic dataset

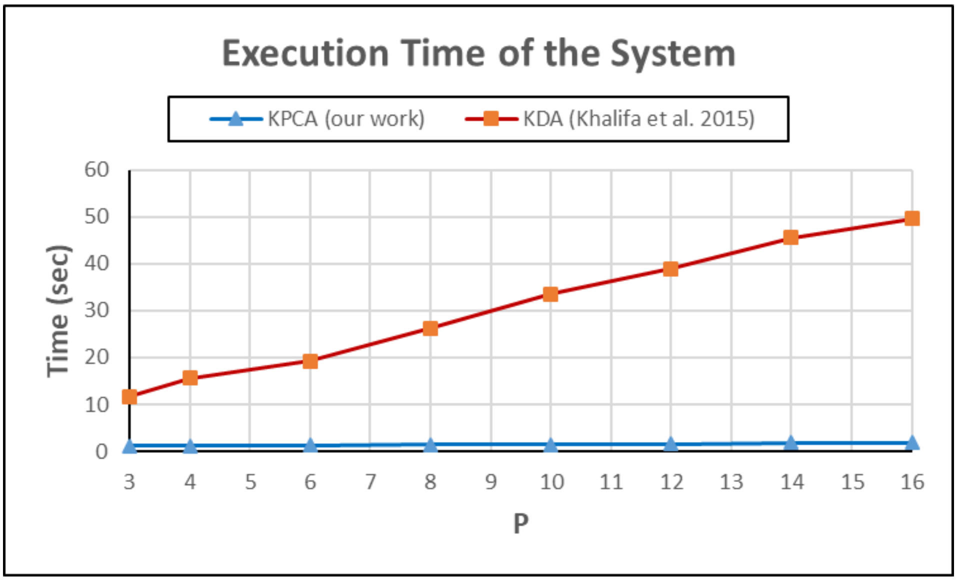



Figure 4.11 Execution time for KPCA vs KDA (Arabic Dataset)

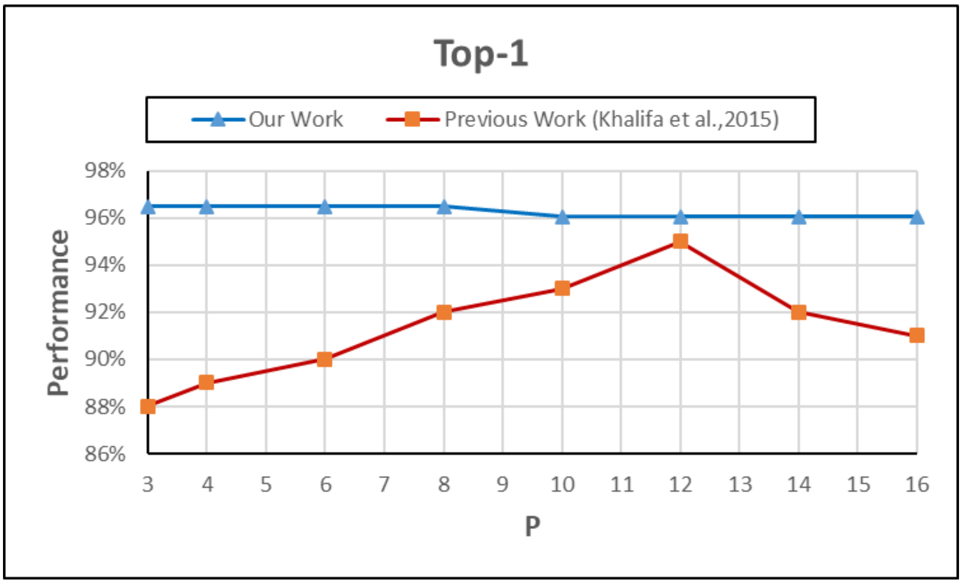



Figure 4. 12 Comparison of the performance for Top-1 (Arabic dataset)

Table (4.5) depicts the system performance in the cases of Top-5 and Top-10 identification rates. In this case, the proposed technique outperforms Khalifa's technique in all cases. The maximum performance of 98.6% is obtained for Top-10 identification compared with 97% for the method proposed by[69]. Therefore, these results clearly show again that our proposed technique yields improved recognition performances. The results of this analysis are illustrated in Figure (4.13) and Figure (4.14).

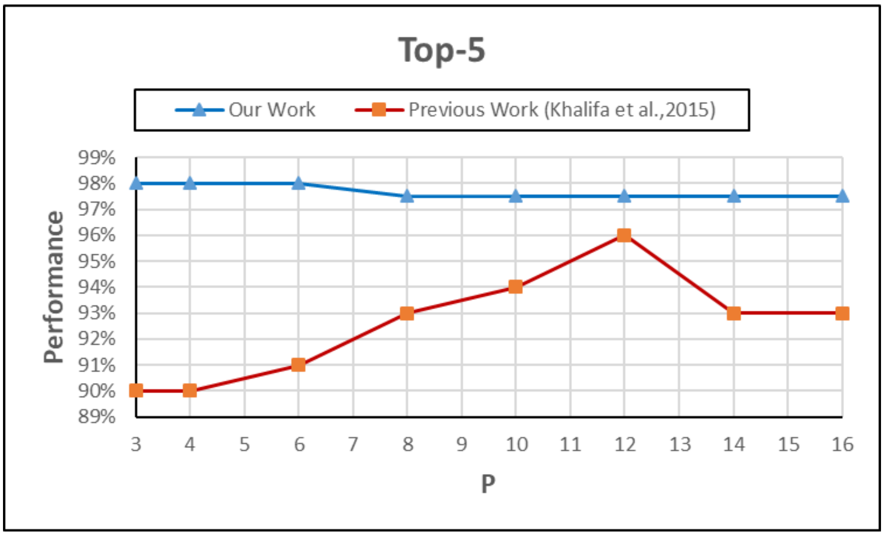



Figure 4.13 Comparison of the performance for Top-5 (Arabic Dataset)

| p | Top-5 (%) | | Top-10 (%) | |
|---|---|---|---|---|
| | **Our work (Based on KPCA)** | **previous work (Based on KDA)** | **Our work (Based on KPCA)** | **previous work (Based on KDA)** |
| **3** | 98.36 | 90.00 | 98.60 | 92.00 |
| **4** | 98.45 | 90.00 | 98.60 | 92.00 |
| **6** | 98.25 | 91.00 | 98.60 | 94.00 |
| **8** | 97.50 | 93.00 | 98.60 | 95.00 |
| **10** | 97.50 | 94.00 | 97.90 | 96.00 |
| **12** | 97.50 | 96.00 | 97.90 | 97.00 |
| **14** | 97.50 | 93.00 | 97.90 | 96.00 |
| **16** | 97.50 | 93.00 | 97.90 | 96.00 |

Table 4.5 System performance for Top-5 and Top-10 for Arabic dataset versus to the performance in [69]

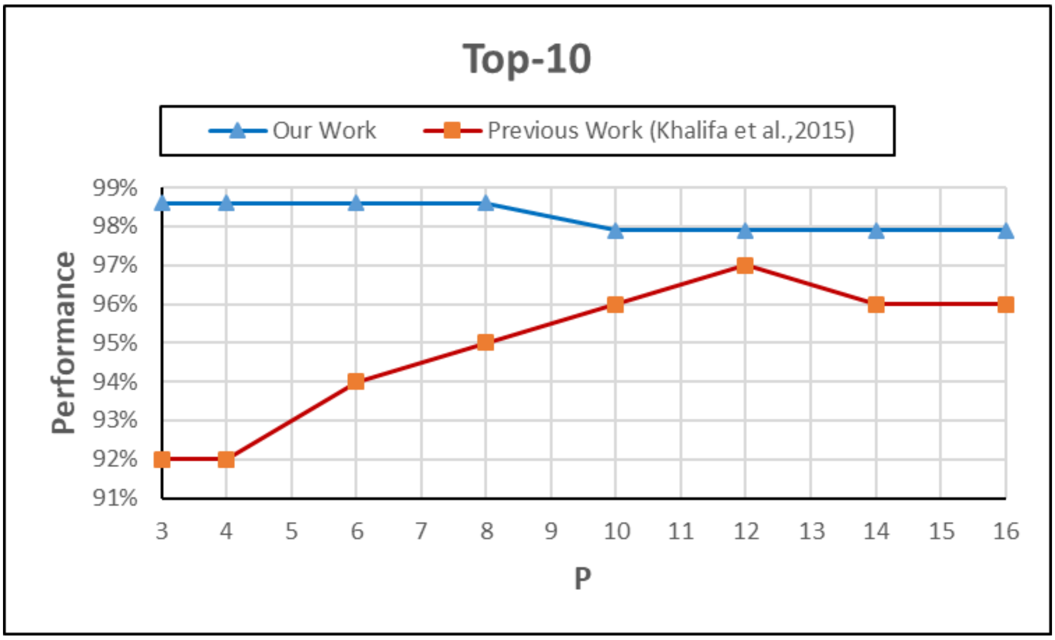



Figure 4.14 Comparison of the performance for Top-10 (Arabic Dataset)

Finally, Table (4.6) shows a comparison of the identification rates between some of the previous works in the literature. It is to be noted that two issues were encountered to perform this comparison: some of the works have used different datasets, which are not available in some cases. Moreover, in the cases where the IAM dataset was used, the number of writers used in the cited works were different; thus, a comparative analysis may not be fair. Therefore, for the sake of a fair comparison, our results have used 650 writers and are compared against other works that have also used the same number of authors. As illustrated in Table (4.6), the results obtained by our proposed method clearly outperform the other listed works that have been developed their systems based on the same numbers of writers.

| Approach developed by | Dataset | No. of writers | (Top1) Performance (%) |
|---|---|---|---|
| Authors [105] | FIREMAKER | 250 | 83.00 |
| Authors [20] | IAM | 150 | 86.00 |
| Authors [105] | IAM | 650 | 89.00 |
| Authors [36] | IAM | 650 | 91.00 |
| Authors [69] | IAM | 650 | 92.00 |
| Authors [69] | ICFHR2012 | 206 | 95.00 |
| **Our Approach** | **IAM** | **650** | **92.00** |
| **Our Approach** | **ICFHR2012** | **206** | **97.00** |

Table 4.6 Comparison of the performance of our proposed approach against similar ones

### 4.5 Summary

This chapter has presented a novel writer identification approach using the concept of Oriented Basic Image feature extraction and its combination with the graphemes codebook method. The proposed algorithm has resulted in an improved identification performance when compared against similar techniques. In addition, the use of KPCA, which is a nonlinear dimensionality reduction technique, has resulted in a reduction of the computational complexity. Further improvement of the identification performance can be achieved by using deep learning (convolutional neural networks) concept. However, the technique is computationally intensive so an implementation using Graphics Processing Unit (GPU) is currently being investigated.

The next chapter aims to further optimise the system performance using various nonlinear dimensionality reduction algorithms such as Kernel Principal Component Analysis (KPCA), Isomap, Locally linear embedding (LLE), Hessian LLE and Laplacian Eigenmaps.

# Chapter 5

# Improving the Performance Through Dimensionality Reduction Technique

## 5.1 Introduction

In the feature extraction phase, the features are extracted from handwritten scripts and are then stored as a very high dimensionality feature vector. This results in a reduction of the system speed while affecting its performance. Therefore, applying dimensionality reduction is a very important step prior to the classification stage. Actually, the dimensionality reduction process can be divided into feature extraction and feature selection. The first approach tries to determine a subset of the original data by using three strategies; filter, wrapper and embedded. The other approach is used to convert the high dimensional data to a space of fewer dimensions. The process can be performed using a linear or nonlinear dimensionality reduction technique. Linear dimensionality reduction techniques are widely used in climate data analysis as a means to support in the interpretation of datasets of high dimensionality [108], while nonlinear dimensionality reduction is used to handle the data generated in the real world [109]. The processing of nonlinear data for further analysis is difficult, but as we will illustrate later, there are many nonlinear dimensionality reduction techniques that can be used to handle this type of data. These nonlinear techniques have the ability to handle nonlinear data and so features conserved help to increase the efficiency of the system [107], [109].

Based on this motivation, this chapter investigates and optimises the performance of an offline English/Arabic writer identification system using a combination of OBI features and the concept of graphemes codebook proposed previously [12]. Therefore, to optimise the system performance, a variety of nonlinear dimensionality reduction techniques such as Kernel Principal Component Analysis (KPCA), Isomap, Locally Linear Embedding (LLE) Hessian LLE and Laplacian Eigenmaps have been used in order to measure and evaluate the best performance value in terms of those techniques. The results obtained have indicated that KPCA is better when compared against similar techniques.

The remainder of this chapter is organised as follows: Section 5.2 discusses the proposed system, Section 5.3, section 5.4, details the experiments carried out to obtain results. Finally, a conclusion is given in Section 5.5.

## 5.2 Proposed Methodology

The offline writer identification system developed in [12] has been designed based on combining a statistical approach and a model-based approach where OBI features are concatenated with codebook of graphemes. For the evaluation, the IAM (English) and ICFHR-2012 (Arabic) databases have been used especially that both datasets have various discriminating features such as handwriting direction, curvature, wrinkliness and distinguishes edge based features which require an efficient feature extraction strategy. On the other hand, to increase the system speed and maintain the system resources, one needs to reduce the very high dimensionality of the resulting feature vector at the testing stage. Therefore, a variety of dimensionality reduction techniques have been utilised, which means, we describe how the performance of the proposed system has been investigated and measured under different nonlinear dimensionality reduction techniques in order to get the optimum value.

At the classification stage, various classifiers were tested, and then Euclidean distance has been adopted as it provides the most attractive results. The experimentations in [12] have been carried out using a single codebook and multiple codebooks, but in this work, we just consider the multiple codebooks. The various stages of this approach are illustrated in figure (5.1). and will be described in the next sections.

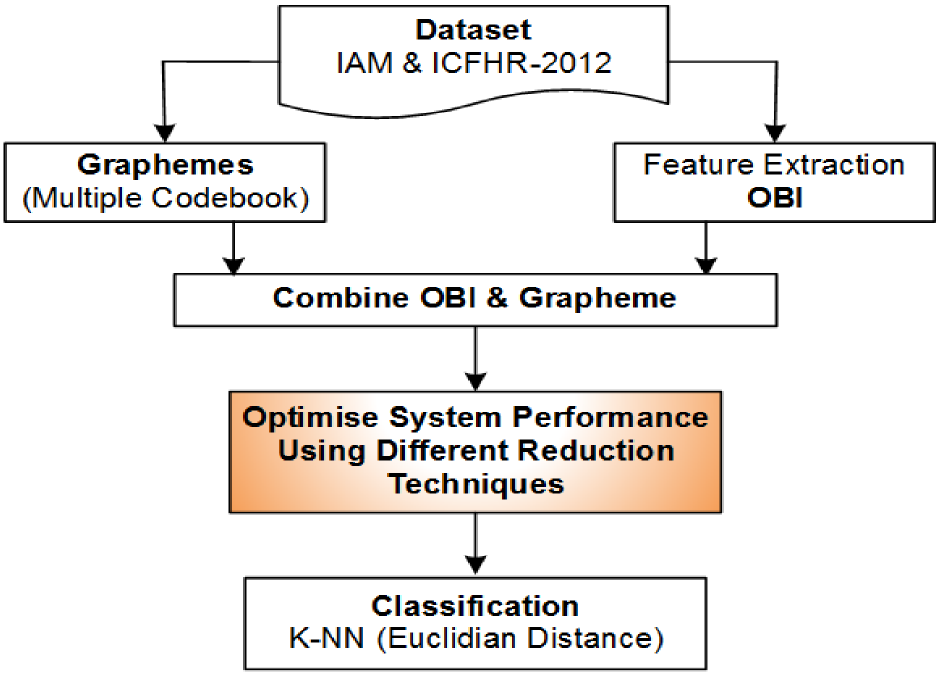



Figure 5.1 The diagram flow of the proposed approach.

### 5.2.1 OBI Concatenation with Grapheme

This method is based on extracting oriented basic image features using a multi-scale approach including local symmetry and orientation. A multi-scale Gaussian derivative filter with different orders and directions is used to generate a number of features and based on the symmetry, orientations, and scales of the features a histogram is created which, after normalisation, generates the final feature vector. Therefore, the proposed approach based on a combination of the OBI feature with a model-based feature (e.g., graphemes-based codebook) is proposed to improve the system recognition performances.

The grapheme codebook method has been proven to be an effective tool in writer identification. The first step of this process is to extract the graphemes which are small pieces or segments of a character. The graphemes can be extracted by chopping the connected components of the written text as necessary and should be achieved based on a suitable deterministic heuristic.

Once done, an image of the dataset should appear as an unordered bag of graphemes which are used to generate the codebook which will act as a reference set of graphemes that will be kept over the whole process in order to determine a 'shape alphabet' with which to describe each dataset image. State-of-art literature shows that there are many codebook generation methods such that the graphemes may be generated from the dataset, can be constructed from a separate training set or they may be computed based on different criteria. Therefore, exist a number of codebook selection methods that are able to create a representative sample of the available graphemes.

In our proposed approach, the dataset in use generates a codebook that is most representative of the data being tested so that the shapes to be recognised will be most closely tuned to the shapes actually used by the authors. We have also used a selection approach to collect the graphemes by a shape-based similarity approach using a Kohonen Self-Organising Feature Map (SOFM) proposed in [106]. The number of clusters has been specified to the required codebook size. Moreover, the cluster centers have been chosen for the codebook and each one is selected as a representative of its cluster of similar graphemes. The SOFM requires extensive training to converge on a layout in order to provide the most effective codebook. Once the codebook has been created, one needs to extract the features. In the case of the single codebook, the performance has been investigated under different codebook sizes of 250, 500 and 1000 using the same codebook size. Using k-means clustering (**k = 250**), (**k = 500**) and (**k = 1000**) allows us to generate multiple codebooks. In this work, we have investigated how system performance is affected when we merge multiple codebook grapheme features. In the first step, the graphemes aggregation of 188k generated in the previous part of the experiment is split randomly into ***P*** (number of partitions) but there is no overlap of graphemes, where ***P = 3, 4, 6, 8, 10, 12, 14, 16***. Therefore, **N** multiple reference base codebooks have been generated from ***P*** and can be used to determine the effect of **N** codebooks on the performance of the identification rate. In the next step, we have combined the OBI feature vector with grapheme features resulting in a new feature vector. The identification performances will be discussed in order to evaluate the usefulness of the proposed method.

### 5.2.2 Dimensionality Reduction Techniques

Different issues in information processing comprise several forms of dimensionality reduction. Dimensionality reduction is an approach that can be used to minimise the number of variables and then the performance of the classification will be improved. Data dimensionality reduction is a critical issue across a number of fields such as pattern recognition, data mining, and so on [109]. Usually, handling high dimensional data produces more complexity through increasing the execution time and memory usage. There are a variety of approaches available that can be used to reduce the dimensions of the data set. Each approach minimise the dimensions based on specific criteria [108].

The main idea of dimensionality reduction is to compact the raw data (high-dimensional data) in a more condensed form by reducing the high dimensionality of the feature vector. This reduces the computational complexity while still keeping intact the accuracy of the recognition [107]. as the algorithm performance will be affected if a noisy or faulty input data have been applied, therefore, removing redundant input data should help the algorithm to achieve higher performances. The main idea of this process has been depicted in Figure (5.2). Moreover, the dimensionality reduction process can be divided into feature selection and feature extraction. The first approach tries to detect a subset of the original data by using three strategies; filter, wrapper and embedded. The other approach is used to convert the high dimensional data to a space of fewer dimensions. Actually, this process can be performed using a linear or nonlinear dimensionality reduction techniques [110]. In this chapter, we specifically consider the problem of the very high dimensionality of the resulting feature vector and how to use nonlinear techniques to handle this vector in order to optimise system performance. Therefore, in this work, we have implemented four nonlinear reduction techniques, which are Kernel Principal Component Analysis (KPCA), Isomap, Spectral Regression Kernel Discriminant Analysis (SRKDA) and Locally linear embedding (LLE). These techniques have been employed to measure the system performance under each case to get the best one.

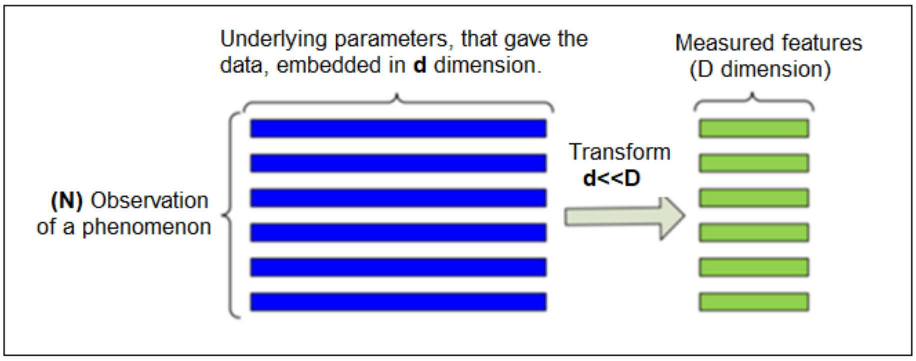



Figure 5.2 The concept of dimensionality reduction

**Kernel Principal Component Analysis (KPCA):** Linear dimensionality reduction techniques are widely used in climate data analysis as a means to support the interpretation of datasets of high dimensionality [108]. These linear methods may not be appropriate for the analysis of data arising from nonlinear processes occurring in a system such as handwriting identification. Therefore, there are many techniques for nonlinear dimensionality reduction which have been developed recently that may provide a potentially useful tool for the identification of low-dimensional manifolds in climate data sets arising from nonlinear dynamics. Linear dimensionality reduction methods have become essential tools for analysing high dimensional noisy data and operate by generating a low-dimensional linear data of the original high-dimensional data while maintaining the important features of the data. Principal component analysis (PCA), which is a very popular linear technique for dimensionality reduction, implements a linear mapping of the data to a lower-dimensional space ensuring that variance of the new data in the low-dimensional is maximised [107], [108] PCA can be used in a nonlinear approach through the kernel trick. Therefore, kernel PCA (KPCA) is the most widely used algorithm for manifold learning [111]. The output method can be employed to construct nonlinear mappings that maximise the variance in the data. The resulting approach is called kernel PCA. The main idea of KPCA works in its core just like PCA, the main difference lies in using a nonlinear mapping which maps each given data point onto an abstract function. In other words, the KPCA technique implements PCA with add some functions, which means, it is a collection of principal component analysis and the kernel trick [107], [109], [112]. At the first step, PCA starts by calculating the covariance matrix of the ( i × j) matrix as shown in the equation (5.1).

$$C = \frac{1}{N}\sum_{i=1}^{N} \mathrm{x}_i \ \mathrm{x}_i^{\mathrm{T}} \qquad (5.1)$$

Where:

- $\mathbf{x}_i \in \mathrm{R}^d$ . *d*-dimensional random vector expressed as a column vector. It operates by diagonalising the covariance matrix $\mathbf{x_i}^{\mathbf{T}}$ .
- Given a data set of ***N*** centered observations in a high-dimensional space.

After that, it evaluates the data onto the first k eigenvectors of that matrix. Here, KPCA starts by calculating the covariance matrix of the data after being converted into a higher-dimensional space as in equation (5.2).

$$C = \frac{1}{N}\sum_{i=1}^{N} \Phi(\mathrm{x}_i) \ \Phi(x_i)^{\mathrm{T}} \qquad (5.2)$$

Where, $\mathbf{\Phi}(\mathbf{x_i})$ is the embedding of data items $\boldsymbol{x_i}$. Let ***F*** is a feature space induced by nonlinear mapping **Φ**: $\mathrm{R}^d \rightarrow \boldsymbol{F}$.

Then it schemes the converted data from the previous step onto the first k eigenvectors of that matrix. It utilises the kernel trick to factor away much of the computation, such that the whole procedure can be executed without evaluating $\Phi(x)$. Obviously, $\Phi$ must be selected and it has a known corresponding kernel.

**Isometric Feature Mapping (Isomap):** Similar to the development process in PCA, Multidimensional scaling (MDS) has been extended in order to implement an approach for nonlinear dimensionality reduction [113]. The developed technique is called Isomap and it is designed based on the MDS approach [113]. Actually, Isomap [110] technique is an integration of the Floyd–Warshall algorithm with classic MDS, which is selecting a matrix of pair-wise distances between all points, and then evaluates a position for each point $xi$. Isomap supposes that the pair-wise distances are only recognised between neighboring points ($xi$ $and$ $xj$), and it utilises the Floyd–Warshall approach to finding the pair-wise distances between all other points. By this way, we can effectively estimate the full matrix of pair-wise geodesic distances between all of the points. Then Isomap technique utilises classic MDS to evaluate the reduced-dimensional positions of all the points.

**Locally linear embedding (LLE):** It is an unsupervised non-linear approach and can be used to test and analyses high-dimensional data spaces and decreases their dimensionalities whereas the local topology must be maintained, i.e. the locations of data that are close in the high-dimensional set must be kept close in the low-dimensional set [113]. This approach has been designed based on the assumption that there is a locally linear relation between contiguous data points. Compared to Isomap, LLE has various advantages, which are faster optimisation and provide better results with many issues. The main idea of this technique is to look in the high-dimensional space and try to represent data item $\boldsymbol{X}_{\mathbf{i}}$ as a weighted combination of its k closest neighbours. This will lead to creating a group of weights $\boldsymbol{w_{ij}}$ for the k neighbours of $\mathbf{X_i}$ and the target is to get a low-dimensional representation $\mathbf{Y_i}$ that should support this weighting. The main function of LLE technique can be represented as following in the equation (5.3):

$$\boldsymbol{\phi}(\mathbf{Y}) = \sum_{i} \left\| \mathbf{y_i} - \sum_{j=1}^{k} \mathbf{w_{ij}} \mathbf{y_{ij}} \right\|^2 \; subject\ to\ \left\| \mathbf{y^{(k)}} \right\|^2 = 1 \quad (5.3)$$

with the sparse weight matrix **W**, the embedding is gained from the $d$ eigenvectors corresponding to the smallest nonzero eigenvalues of $(\mathbf{I} - \mathbf{W})^{\mathbf{T}}(\mathbf{I} - \mathbf{W})$

<u>**Hessian LLE:**</u> Hessian LLE is developed to use the same concept of LLE. It reduces the curvature of the high-dimensional manifold when finding the low-dimensional representation [113]. The process performs local isometry between the distances in both spaces. Here we need to apply PCA to every data point $\boldsymbol{xi}$ and its k nearest neighbors, so this will produce an approximation of the local tangent space at every data point. Then we can get the mapping function **M** from the **d** principal components for every point $\boldsymbol{xi}$. Then this mapping function can be utilised to yield an estimator for the $d(d + 1)/2$ - dimensional Hessian $\boldsymbol{Hi}$ of the manifold at that data point.

We suppose the setting where our sampled data is ($\boldsymbol{m_i}$). The input data: ($m_i$ : i = 1, . . . , N) a collection of N points in $\mathbf{R^n}$. Also, the parameters; **d:** dimension of the parameter space; **k**, the size of the neighbourhoods for fitting. From the Hessian estimators in tangent space, a matrix $\boldsymbol{H}$ is constructed with entries as equation (5.4).

$$\mathrm{H_{i,j}} = \sum_{l} \sum_{r} ((H^l)_{r,i} (H^l)_{r,j}) \qquad (5.4)$$

Where $\boldsymbol{H^l}$ , again, the d (d+ 1)/2 **k** matrix associated with estimating the Hessian over neighbourhood $\mathbf{N}^{\boldsymbol{l}}$, where rows $\boldsymbol{r}$ corresponds to specific entries in the Hessian matrix and columns $\boldsymbol{i}$ correspond to specific points in the neighbourhood.

The eigenvectors that match to the d smallest eigenvectors of **H** can be utilised to determine the low dimensional embedding **Y** that reduces the curvature of the manifold.

<u>**Laplacian Eigenmaps**:</u> Laplacian Eigenmaps utilise spectral approaches to carry out dimensionality reduction. It aims to detect a manifold representation that can be used to maintain a group of similarities in a local neighborhood for the high-dimensional data[113]. Weights $\boldsymbol{w_{ij}}$ are specified as the similarities between subjects within a local neighborhood. For all other pairings, the weights are given zero. Actually, the similarities came from distances $\boldsymbol{d_{ij}}$ using a heat kernel such as equation (5.5).

$$w_{ij} = e^{-\frac{d_{ij}^2}{t}} \qquad (5.5)$$

where *t* is the kernel width. The Laplacian eigenmaps embedding is gotten by reducing the objective function in equation (5.6).

$$\phi(Y) = \sum_{ij} \left\| y_i - y_j \right\|^2 w_{ij} = 2Y^T LY \qquad (5.6)$$

where **L = D – W** is defined as the graph Laplacian matrix that has been derived from the weight matrix **W** and the diagonal degree matrix **D** where $\boldsymbol{D_{ii}} = \sum_{\boldsymbol{j}} \boldsymbol{W_{ij}}$.

## 5.3 Classification

In our previous system [12], we have investigated Euclidean distance, Chi-square distances and Manhattan distance and the results obtained are as follows: Manhattan distance = 80%, Euclidean distance = 96.05%, and Chi-square distance = 73%. This has allowed us to adopt the Euclidean distance in our work since it outperforms the other metrics.

## 5.4 Experiments and Results

In this section, we present the experiments conducted in order to measure and evaluate the system performance with some of the dimensionality reduction techniques as mentioned before. The codebook is generated and used in the system for English and Arabic datasets. Therefore, to show the effectiveness of the dimensionality reduction, the feature vector, which represents the high dimensional data in this work, is processed by a group of the dimensionality reduction techniques. In this experiment, we present the evaluation of the system performance by employing multiple codebooks and processed them against each one of the reduction techniques described. This means, in the implementation steps, the datasets (English and Arabic) were first pre-processed by using one of the dimension reduction techniques and then classification has been performed. The results obtained from the experiments will be compared in order to find the optimum value of the system performance. For an objective comparison, in each experiment. the performance of each one of the selected reduction techniques has been investigated using the same classifier, which is Euclidean distance as mentioned above. We developed our codes of the dimensionality reduction techniques that we selected in this work based on Matlab Toolbox for Dimensionality Reduction [114].

### 5.4.1 Measure of the performance using English Dataset

In this experiment, we use the high dimensionality data, which is represented by the features vector, versus some of dimensionality reduction techniques and then the system performance is measured, and the results are shown in the Table (5.1). The obtained results are attractive. The system performance for Top-1 is evaluated versus each one of the reduction techniques described above, which means the results present based on different values of ***P*** (Codebook partition). the system performance increases versus P in all techniques as expected. KPCA and H.LLE slightly outperform the others with an improvement of about 5% between the first and last cases. However, in most cases that implemented against different values of P; while KPCA and H.LLE have the best performance with P=16, but the improvement is limited.

According to the obtained result in Table (5.1), both KPCA and H.LLE generated the average of performance more than 88.8%, which is most precise in verifying the data. Moreover, this result proves that both techniques have fulfilled the target in obtaining the individuality of handwriting and also were implemented out successfully to minimise the feature to a new space with low-dimensional data. That means the converted features successfully represent the distinction of the writer, so in this case, the writer identification process will be easier, and the developed system can use the resources in an efficient manner and that will lead to improvement of the system performance.

| | **Top (1) - System Performance (%)** | | | | |
|---|---|---|---|---|---|
| **p** | **KPCA** | **ISOMAP** | **LLE** | **H. LLE** | **Laplacian** |
| **3** | 86.00 | 80.32 | 84.37 | 85.40 | 78.54 |
| **4** | 87.00 | 81.54 | 85.36 | 85.22 | 80.09 |
| **6** | 89.00 | 83.91 | 87.33 | 88.24 | 84.29 |
| **8** | 89.95 | 82.40 | 86.62 | 88.98 | 81.24 |
| **10** | 92.03 | 85.94 | 90.19 | 92.82 | 83.98 |
| **12** | 89.57 | 86.83 | 87.36 | 87.93 | 87.34 |
| **14** | 90.25 | 83.91 | 87.85 | 88.54 | 84.34 |
| **16** | 90.41 | 86.39 | 88.67 | 89.70 | 85.44 |

Table 5.1 Measure System Performance Based on Different Dimension Reduction Techniques - English Dataset

Figure (5.3) illustrates how the system performance change according to the codebook partition ***P*** when each dimensionality reduction technique that was mentioned above is used. From this figure, for most cases, we can see the system performance is gradually improved as the codebook partition ***P*** increases. In other words, increasing ***P*** leads to improved system performance.

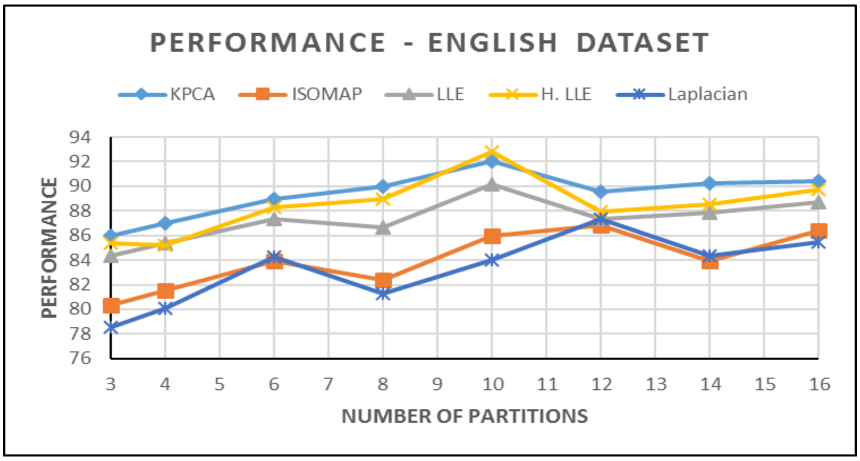



Figure 5.3 System Performance for English Dataset

### 5.4.2 Measure of the performance using Arabic Dataset

The results in the Arabic dataset show a similar pattern. That is, all dimension reduction techniques followed by the same value of ***P*** give performance shown in Table (5.2). Again, in the case of the Arabic dataset, the system performance has been found and investigated for Top-1 versus each one of the reduction techniques described above. Also, the similar result in the case of English dataset is obtained here, which means, for each reduction technique, the value of the system performance increases versus ***P*** as expected, and again KPCA and H.LLE techniques slightly outperform the others with an improvement of about 1% between the first and last case. Moreover, we have observed that the performance is approximately constant in the case of KPCA because the distance that evaluated by Euclidian distance is almost fixed, which means there is no effect of increasing ***P***. Figure (5.4). illustrates how the system performance change according to the codebook partition ***P*** when each dimensionality reduction technique mentioned above is used.

| p | Top (1) - System Performance (%) | | | | |
|---|---|---|---|---|---|
| | **KPCA** | **ISOMAP** | **LLE** | **H. LLE** | **Laplacian** |
| **3** | 96.49 | 89.34 | 92.45 | 94.34 | 84.37 |
| **4** | 96.49 | 89.87 | 92.89 | 94.67 | 84.23 |
| **6** | 96.49 | 89.13 | 92.17 | 94.77 | 85.9 |
| **8** | 96.49 | 90.45 | 93.56 | 95.16 | 85.12 |
| **10** | 96.05 | 90.25 | 93.87 | 94.89 | 85.38 |
| **12** | 96.05 | 90.85 | 94.84 | 95.56 | 86.67 |
| **14** | 96.05 | 90.38 | 94.33 | 95.05 | 86.78 |
| **16** | 96.05 | 88.45 | 94.13 | 94.13 | 85.15 |

Table 5.2 Measure system performance based on different dimension reduction techniques - Arabic Dataset

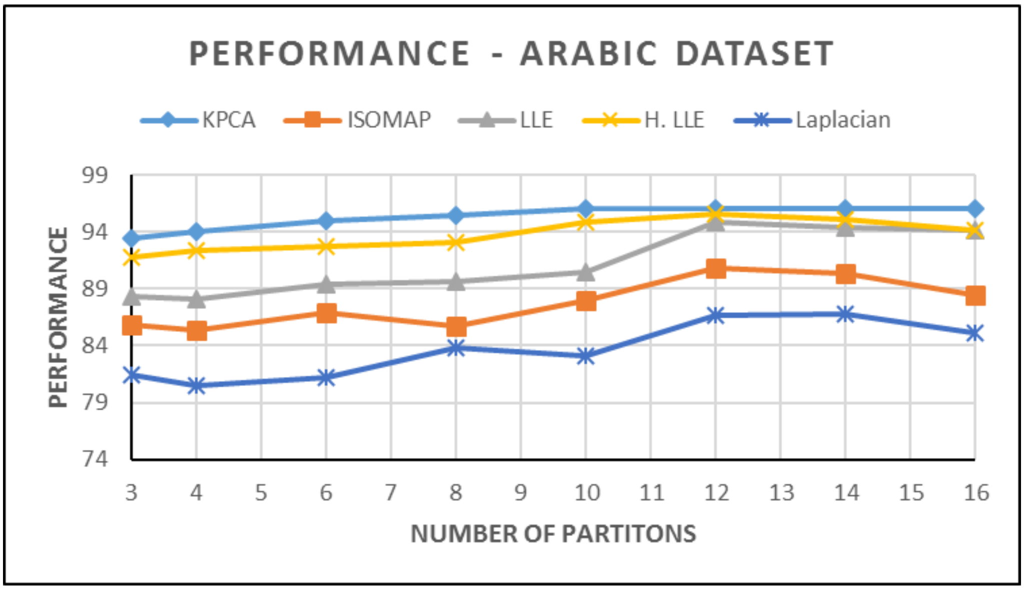



Figure 5.4 System Performance for Arabic Dataset

## 5.5 Summary

Dimensionality reduction can be defined as a process that can be used to extract the fundamental information from the high-dimensional data set. In this work, we present some approaches to reduce the dimensions of the raw data (features vector) that is represented by the output of the feature extraction stage. Therefore, several nonlinear dimensionality reduction techniques have been investigated to get the optimum value of the system performance. The low dimensional data created from reduction techniques are again processed by using Euclidean distance classification. The obtained results show that the KPCA method has provided the best result compared to other reduction techniques used in this work. Also, the results proved that measuring and optimising the system performance of the writing identification based on dimensionality reduction techniques plays an important role of the writer recognition and handwriting classification.

The next chapter aims to assess and compare the performance between one of the deep learning algorithms, the AlexNet model, with two of the most effective machine learning classification approaches: Support Vector Machine (SVM) and K-Nearest-Neighbour (KNN). The evaluation has been conducted using both IAM dataset for English handwriting and the ICFHR 2012 dataset for Arabic handwriting.

# Chapter 6

# Writer Identification Using Machine Learning Approaches: A Comparative Study

## 6.1 Introduction

The performance of writer identification highly depends on the selected features and the used classifier. Therefore, robust and efficient approaches are required to determine and extract the most discriminative features. Due to the extensive variation of the data from a writer to another, traditional machine learning approaches are not able to provide the most reliable identification performances. Recently, Deep Learning (DL) has emerged as an attractive concept capable to analyse complex tasks thus enabling much higher performances. This chapter aims to investigate the power of DL for offline writer identification. To achieve this, we aim to carry out a comparative study of the performances achieved against two well-known and proven machine learning algorithms to demonstrate the usefulness of DL.

Deep learning, which was proposed in 2006 by Geoffrey Hinton, has become a powerful method to deal with various computer vision and pattern recognition problems [115]. The main idea of the DL approach is to use several hidden layers of a neural network in order to create a more effective high-level representation of the data by using a group of low-level features. Moreover, deep neural network (DNN) can be implemented using various architectures including Convolutional Neural Networks

(CNNs), Autoencoders, Deep Belief Nets (DBN), Recurrent Neural Networks (RNN) and more [116], [117].

CNN's, which are one of the most common DNN architectures, use convolutional layers with pooling layers that reflect the translation-invariant nature of the processed images. In fact, a CNN architecture can be seen as a powerful model of deep learning and has been successfully used to recognise and classify images effectively with various distortions [115], [117], [118]. As shown in Figure (6.1), this chapter aims to investigate DL for writer identification task and its evaluation through its analysis against traditional machine learning. To achieve this, AlexNet architecture, which is one of CNN models, has been selected and compared against two of the most effective machine learning classification approaches; the SVM and KNN. For traditional machine learning approaches, a combination of Bag of Words (BoW) extracted using SIFT and SURF with SVM and KNN as classifiers. Typically, AlexNet is used as an automated identification process and can be used to select, extract features and to construct new ones. All the experiments have been conducted based on the IAM and ICFHR-2012 datasets that we used before in [12], in order to compare with this work.

This chapter is organised as follows. Section 6.2 gives an overview of CNN and we tried to cover the topics that related to writer identification. The main idea of the machine learning workflow is presented in section 6.3. Then section 6.4 describes writer Identification using the AlexNet model. The dataset used in this work is shown in section 6.5. The experiments and results are presented in sections 6.6 and 6.7. The conclusion of this work has been presented in section 6.8.

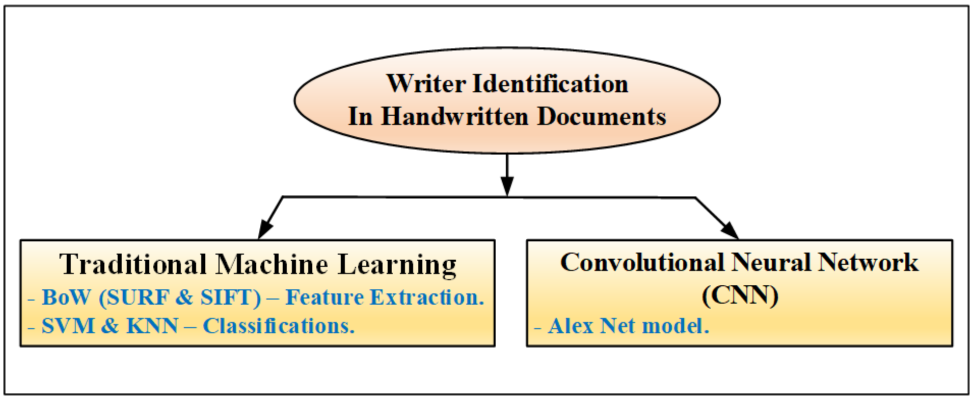



Figure 6.1 An approach for performing offline writer identification

## 6.2 Overview of CNNs

The motivation of this work is to investigate reliability for writer identification tasks throughout a comparison between some of the machine learning techniques and deep learning-based algorithms in order to prove how CNN models can be used to manage complex issues in the computer vision area. This section provides the fundamental knowledge and the state-of-the-art methods about deep learning in the topic of the writer identification process. In recent years, the use of deep learning has increased interest in computer vision applications and especially in writer identification [12], [119], [120]. Many studies have used different models of deep neural network which is the most effective and supervised machine learning approach. Due to the importance of writer identification as a significant task for the automatic processing of handwritten documents, several attempts have been made to automatically identify the writer in the past few years. Prior to the emergence of deep learning techniques, most of the handwriting identification studies have been done based on SVM as a classifier, but some image features have to be extracted first.

Some distinguished works have been carried out in [12] and [94] where several statistical and model-based features were proposed. The authors in [94] proposed an approach to improve a statistical feature and the edge hinge distribution by exploring a combination of the features with a codebook of graphemes generated using a model-based feature extraction approach. A writer identification method was presented in [12] using the concept of Oriented Basic Image feature extraction and its combination with the grapheme codebook method. In addition, numerous researches have attempted to evaluate several feature extraction approaches and learning algorithms in order to enhance the performances of writer identification [1], [3], [94], [115]. However, it was very difficult to make a comparison between these studies because of the various and large feature extraction methods including different types of datasets used in this researches.

During the past few years, a variety of sophisticated methods based on deep learning have been used in writer identification issue. However, each has its advantages and drawbacks, but the concept was used to obtain the best deep learning architecture. Numerous studies using CNNs have been widely used by several researchers as an effective approach for tasks such as document analysis, image classification and object recognition. In most cases, DCNNs can be trained on large datasets to achieve high performance resulting in very high computation of the which is a big issue for such networks. The authors in [120] use DCNN as generic feature extractors and a previously trained network was reused in order to manage this problem. At the start, a DCNN model was trained on the ILSVRC-12 dataset. Then, the learned neurons (kernels) are preserved and only the classification section is retrained based on different datasets. A similar strategy to overcome this issue has been used in [121] for the task of object recognition. The authors used pre-trained CNN with pre-processed depth images. The features extracted by the CNN and then used by SVM in order to determine the objects.

Their proposed model has been evaluated using the Washington RGB-D Objects dataset.

Offline writer identification can be classified into two levels of analysis: the texture-based and allograph-based. The latter approach depends on local descriptors calculated from tiny letter pieces (allographs). In the same context, the authors in [119] introduced a method for writer identification using the features extracted from CNNs as local descriptors. A global descriptor was created through GMM super vector encoding. Also, the authors compared their proposed method to traditional local descriptors such as SIFT and SURF. The proposed approach is evaluated using two datasets; the ICDAR 2013 and the CVL. Another interesting method for CNN is the model proposed in [122] where a CNN architecture is employed for writer identification and retrieval. A feature vector is created for each writer and the identification is carried out using a dataset containing pre-calculated feature vectors. Their approach has been evaluated using ICDAR2013, ICDAR 2011 and CVL datasets.

## 6.3 Machine Learning Workflow

In the case of writer identification, a machine learning algorithm can be used to assign a handwriting script sample to one author among many available in the dataset [1], [12]. At the learning stage, the algorithm extracts discriminative features from several writers and generates a feature vector for each writer used. Then, given a handwritten sample, the algorithm will extract a feature vector and compares it against all learned feature vectors of the learning stage. Using distance metrics, the closest feature vector will identify the more likely writer. Various feature extraction methods with various classifiers such as SVM and KNN can be used [12], [120].

In this chapter, the concept of BoW has been adopted as a feature extraction phase. Therefore, SURF and SIFT have been selected as extraction methods in the BoW model and then are combined with SVM and KNN as classifiers. This proposed approach has been adopted due to their effective performances as demonstrated in writer identification. Usually, the concept of BoW consisting of three main steps: feature detection, feature description, and codebook generation.

• **Feature Detection**: Feature detection is an operation used to process an image at low-level to determine low level features

• **Feature Description**: After feature detection step, each image is abstracted by several local patches which are then represented as numerical vectors called feature descriptors. A descriptor is usually able to process intensity, rotation, scale and affine variations to some extent. As mentioned above, in this work, SIFT and SURF, which are effective extraction techniques have been used as a feature descriptor to extract distinguished features and descriptors from images.

• **Codebook Generation**: In the last step, the BoW model has been utilised to convert vector-represented patches to the codebook.

Then at the last stage and as mentioned above, we use well known classifiers SVM and KNN in order to classify the test images.

### 6.4 Writer Identification using CNNs

A CNN architecture has been adopted and comprises of five different layers: input, convolution, pooling, fully-connected and output layer [123]. The image input layer specifies the size of the input images. Then in the next layer, the input patch is convolved based on different learned kernels using shared weights. Next, the pooling layer has been tried to minimise the size of the image while trying to preserve the basic information. This process can be achieved by implementing a max pooling over image parts of size 2x2 or 3x3 [120]. By performing these two layers, the feature extraction phase has been composed. After that, the extracted features are processed by allocating weighting values and then merged in the fully connected layer. This step represents the classification phase of the convolution network [119], [120]. However, to make CNN work very efficiently and provide improved results, it is essential to use a large dataset. Therefore, in this chapter, a well-known and used CNN model called AlexNet with trained datasets has been used in order to tackle the problem of high computation of the training phase. The first part is used to extract features while the classification part is performed based on SoftMax. In addition, a very efficient GPU, which is based on NYIDIA GeForce GTX 770 with 16 GB of RAM, has been used in order to make training faster during the convolution operation. Although, there exist various CNN architectures [123], [124] including LeNet, ZF Net, GoogLeNet, VGGNet and ResNet (2015). AlexNet has been chosen comprising a number of hidden layers: an input layer, five convolutional layers, three pooling layers, three fully-connected layers, and finally an output layer. The main advantages of the model are the ability to attain outstanding performance with reduced training parameters and a strong qualitative robustness.

However, The Alex-Net model requires the selection of convenient approaches in order to make the training process faster and to avoid over-fitting caused by complicated structure, a large number of training parameters and a vast amount of data. Consequently, the model has been constructed by inserting the Rectified Linear Unit (ReLU) in the data structures. In addition, the dropout approach has been employed in the fully connected layers (FC6, FC7, and FC8) to enhance the robustness so that the learning process of the hidden layers remains independent on the extracted features from the upper layer [125].

## 6.5 Datasets used

The developed system has been evaluated using the IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting. IAM dataset is one of the most popular English handwritten datasets for writer identification and verification. The IAM test consists of a total of 1314 handwritten samples with two samples per writer. The training part contains only the third and fourth samples. These samples have been provided from 127 writers and each writer provided at least four samples. On the other hand, the data used for testing is taken from the 1st and 2nd samples of all the 657 writers. ICFHR 2012 dataset uses the Arabic language. The documents were scanned and provided as greyscale PNG images of varied contents of the text. Over 200 writers were requested to write three varied content paragraphs in Arabic.

## 6.6 Experiments

In this section, we present the main steps of the work, which have been designed based on the concept of pattern recognition task for the writer identification process. There are two main phases; feature extraction and then select the suitable classifier. As mentioned above, this work aims to investigate reliability for writer identification tasks throughout a comparison between machine learning and deep learning techniques. The work has been developed by using the Matlab Environment. In the next sections, the feature extraction algorithms have been listed and types of classifiers and their parameter selection.

### 6.6.1 Feature Selection and Extraction

In this phase, we have used two techniques to obtain a feature vector for each document image (script) in the dataset. The first one was to utilise hand crafted features of an image, which means machine learning techniques such as SURF and bag of words. The other technique was to use pre-trained CNN such as the AlexNet model. Given below is the brief description of feature vectors that have been used in this work.

- **Bag-of-Words (BoW):** BoW model is a simple algorithm used in the computer vision field and known as Bag-of-Features (BoF) or Bag-of-Visual Words (BoVW) [126]. The main idea of this approach is to start by selecting an input image. After that, there are some algorithms such as the Harris-Laplace corner or dense sampling strategy can be used to find the interest points and also referred to as detectors. They are usually used to choose local image patches. Then, when the image patch has been detected, we have to use one of the image descriptors, in this experiment, SIFT has been selected as an image descriptor in order to extract visual information. Finally, the low-level features that have been obtained will be transformed by using the BoW approach.
- **SURF:** SURF has proven to be a very effective algorithm in object detection and pattern recognition applications. But the main disadvantage that it needs a large computational complexity when used for real-time applications [58]. However, it is used in several applications, because of its powerful attributes such as scale invariance, translation invariance, lighting invariance, contrast invariance, and rotation invariance. Moreover, the algorithm can be used to detect objects in images taken under different extrinsic and intrinsic conditions.

The algorithm has been designed based on four main parts as shown in Figure (6.2):

1) Integral image generation,

2) Fast-Hessian detector (interest point detection),

3) Descriptor orientation assignment (optional),

4) Descriptor generation.

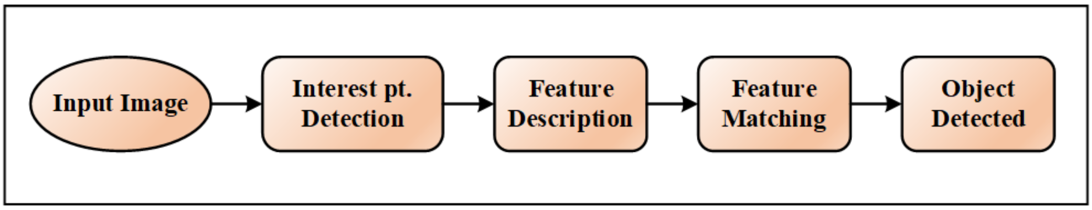


Figure 6.2 Flow of SURF algorithm

- **AlexNet Model:** AlexNet model is the second technique that has been used to extract features. It is a pretrained CNN model and has provided an efficient approach to get the feature vector for both training and test dataset. To obtain the features using the AlexNet model, the fully connected layers FC6, FC7 and FC8 are used to get the learned feature vector of size 4096, 4096 and 1000 neurons respectively.

### 6.6.2 Classification Methods

The classification phase is essential because it is used to allocate a sample (or a form) to predefined class based on a relation between the data processed objects with a pre-defined class label[61]. In this work, we select two of the most efficient classifiers, which are, KNN and SVM. They have been used under machine learning techniques. On the other hand, the softmax layer in AlexNet model has been used as a classifier.

- **KNN:** this technique uses a vector space model in order to classify data points [57] and depends on the fact that the data within the same class will have high similarity. Therefore, when we need to calculate the similarity with the known class, the class of unknown class can be estimated. Since the performance of the whole process will be changed and impacted with the choice of a number of neighbours ($k$). To improve performance, we need to examine multiple $k$ and then select $k$ to increase test accuracy and also reduce over fitting [127]. More details about KNN have been covered in Chapter (2).
- **SVM:** it conducts classification tasks through constructing hyperplanes in a multidimensional space that can be used to split cases of several class labels. SVM approach is a supervised learning model and usually supports regression and classification tasks and can be used to address multiple continuous and categorical variables [127]. More details about SVM have been covered in Chapter (2).
- **Softmax:** Softmax is the output layer of the AlexNet model and used as a classifier. It looks at the output of the previous layer, which represents the activation maps of high-level features, and determines which features most correlate to a particular class.

Therefore, as shown in Figure (6.3), this work has investigated seven different scenarios for constructing the classifier.

### 6.7 Results

In this section, a collection of experiments and results are presented and discussed. As mentioned above, there are two main parts of experiments: the first part of experiments includes evaluating the performance based on machine learning algorithms; a bag of words technique and SURF for feature extraction and then using SVM and KNN for classification. The second part of the experiments investigates the system performance using the AlexNet model. All the experiments have been carried out on the datasets mentioned above.

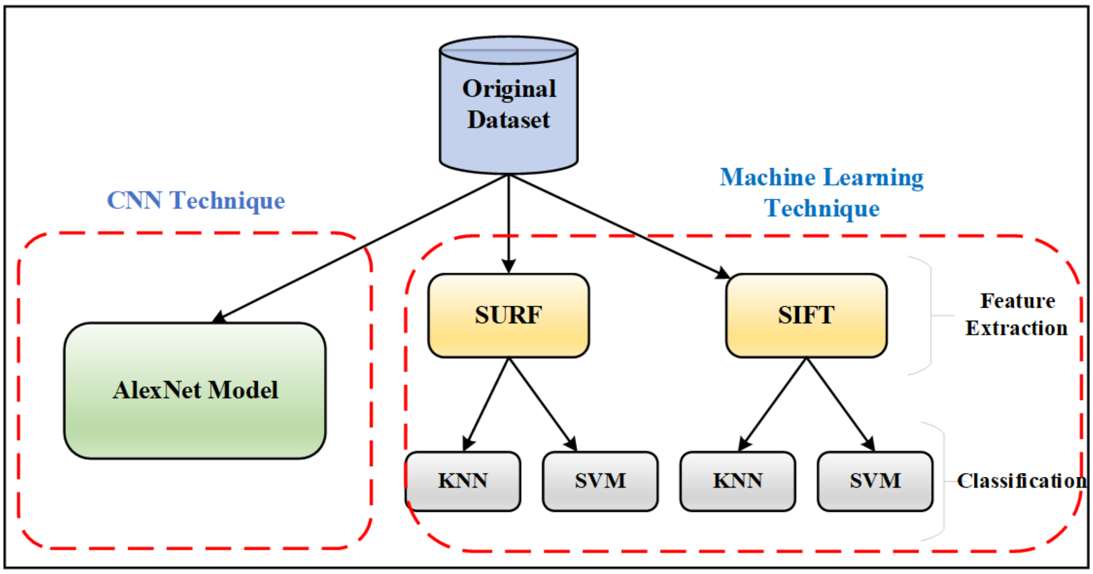



Figure 6.3 The Experiment Scheme

### 6.7.1 Evaluation of Performance based on Machine Learning Model

In this experiment, the image classification analysis has been carried out using SURF and SIFT descriptors where the classification employs SVM and KNN in order to classify the test images. The identification performance has been evaluated and shown in Table (6.1) and Table (6.2) based on our datasets IAM and ICFHR-2012. The obtained results as shown in Figure (6.4) and Figure (6.5) indicate that it easy to observe that the SVM classifier in both cases (SIFT and SURF) outperforms slightly the KNN classifier.

| Classifier | KNN (%) | SVM (%) |
|---|---|---|
| IAM | 90.0 | 90.5 |
| ICFHR-2012 | 94.0 | 95.0 |

Table 6.1 System Performance (Top-1) based on SURF

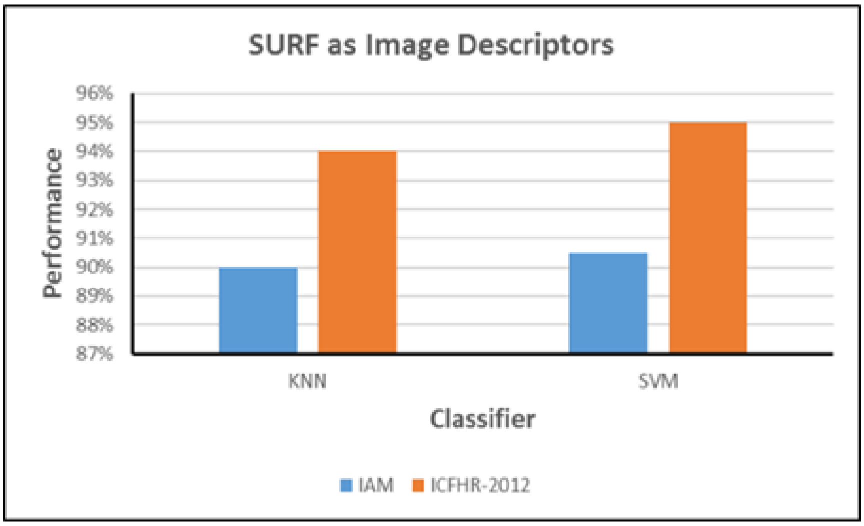



Figure 6.4 Performance of the machine learning model based on SURF

| Classifier | KNN (%) | SVM (%) |
|---|---|---|
| IAM | 89.0 | 90.0 |
| ICFHR-2012 | 93.0 | 94.0 |

Table 6.2 System Performance (Top-1) based on SIFT

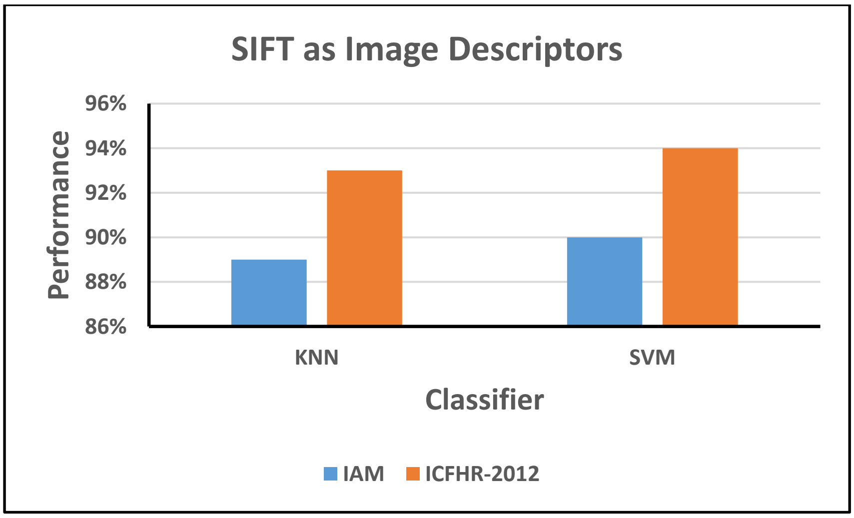



Figure 6.5 Performance of the machine learning model based on a bag of words

### 6.7.2 Deep Learning

In order to assess DCNN approach using AlexNet model, the identification performance is evaluated by using the image features that are extracted from fully from the connected layers of Alex-Net model (FC6, FC7 or FC8). Then, we determine which fully connected layer provides the best identification performance. In this part, a softmax classifier has been used. In the first step, we extract the image features from one of the fully connected layers in the AlexNet model FC6, FC7 or FC8 (one at each time). Then, these image features have been classified by using softmax. These features are examined individually with respect to their respective performances. The identification performance of our proposed approach is shown in Table (6.3). The obtained results as shown in Figure (6.6) indicate that the best performance is provided from FC7, but in the next layer, FC8, the recognition accuracy has fallen. That happened, because the higher layer FC8 of the Alex-Net model maybe cannot recognise much between the features because they capture only the abstract and high-level information, while the mid-level features in the FC7 layer have additional distinguished power for same-class recognition.

To simplify the comparison between two approaches above, the average performance of the CNN model (AlexNet) has been compared against two machine learning approaches; SVM and KNN. Although, we noticed that SVM and KNN achieved good results, but the CNN model produced better results over the same datasets.

**6.8 Summary**

In this chapter, the writer identification performances of machine learning algorithms using SURF and SIFT approaches for feature extraction and SVM and KNN classifiers have been compared against deep learning algorithms such as CNN (Alex-net model). The analysis was carried out using a comparative study using the IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting. These two datasets are widely used databases for writer identification. The obtained results show that deep learning outperforms machine learning algorithms.

| Dataset | FC6 (%) | FC7 (%) | FC8 (%) | Average (%) |
|---|---|---|---|---|
| IAM | 90.0 | 91.0 | 90.3 | 90.4 |
| ICFHR-2012 | 96.6 | 97.5 | 93.6 | 95.9 |

Table 6.3 Identification performance based on FC layers

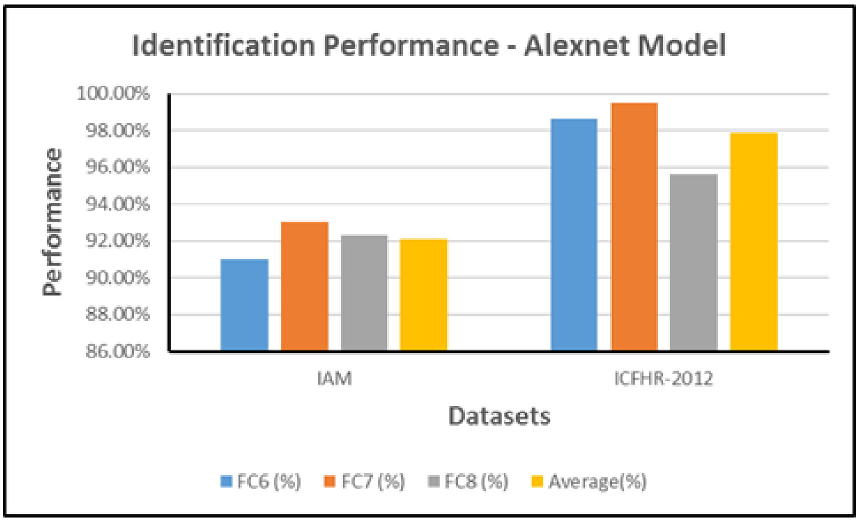



Figure 6.6 Identification performance using AlexNet model

The next chapter presents an efficient approach for handwriter identification using deep learning technology. A deep learning network has been used to capture and learn intrinsic writer features to build discriminative feature vectors associated with each writer. A machine learning algorithm is then used to classify the features and hence identifiy the corresponding writer. This means the Alex-Net model has been used to extract image features from the fully connected layers then classified using an SVM classifier to effectively improve the identification performances.

# Chapter 7

# Writer Identification using Deep CNN Architecture

## 7.1 Introduction

This chapter presents a method using the concept of deep learning to further improve the performances. Deep learning (DL) is a relatively new approach to machine learning and has become a powerful research tool especially useful in various computer vision and visual problems [128], [129]. The main idea of DL is to use a number of hidden layers of a neural network in order to create a clear abstract high-level representation of the data using a group of low-level features. Moreover, deep neural networks (DNNs) have a number of different architectures, such as convolutional neural networks (CNNs), autoencoders, deep belief nets (DBNs), recurrent neural networks (RNNs) and others [117]. Among them, CNN architecture is a powerful model and has been successfully used to recognise and classify images in many problems [115], [117]. The efficiency of CNNs has been proven in various problems including computer vision and image classification [130], [131], object recognition and detection [132], face recognition [133] , and handwriting recognition in different languages.

In this chapter, the proposed approach is developed based on our previous work [134], which has been aimed to investigate CNNs for writer identification method and its evaluation through its analysis against conventional machine learning. AlexNet architecture has been selected and compared against a few machine learning techniques such as Support Vector Machine (SVM) and K-Nearest-Neighbour (KNN) as described in [134]. Although conventional machine learning offers flexibility in terms of providing various feature extraction and classifier methods, deep learning methods have the ability to learn a very large number of discriminative and intrinsic patterns thus making it possible to attain very high performances. However, this comes at the expense of the necessity of required very large data for the training stage thus making is computationally intensive. Therefore, selecting a suitable feature extractor or set of feature extractors has the potential of further improving the identification accuracy. Therefore, the main contribution of this chapter is to concatenate CNN (Alex-Net model) with machine learning techniques in an efficient way. In order to deal with this challenge, there are two steps:

1. The first one uses Alex-Net as a powerful feature extractor in order to efficiently extract features from one of the Alex-Net fully connected layers (FC6, FC7 or FC8). This can be seen as a design strategy.
2. In the second step, the softmax, which is the last layer of the Alex-Net model and it is mainly used as a cost function for probabilistic multi-class classification, is replaced by one classifier to enhance the identification performances. In this chapter, the SVM classifier [22] has been selected especially due to its hinge-loss cost function which results in a maximum margin hyperplane. This can be extended to non-linearly separable problems using kernel approaches by mapping the data into higher dimensional spaces.

Figure (7.1) shows how DL has been used in our work as a feature extractor for the writer identification application. In the first phase of the process, a classical Alex-Net model has been used to extract image features from fully connected layers of the FC6, FC7 or FC8 models to provide 4096, 4096 and 1000-dimensional feature vectors, respectively. Then, in the second phase, the SVM classifier is applied

The rest of the chapter is organised as follows. Section 7.2 illustrates an overview of extracting features and classification using CNN. Section 7.3 briefly illustrates the structure of the convolutional neural network model. Our proposed approach is then presented in section 7.4 while the data sets used in the evaluation are presented in section 7.5. The experiments and results are shown in section 7.6 and the conclusions are drawn in section 7.7.

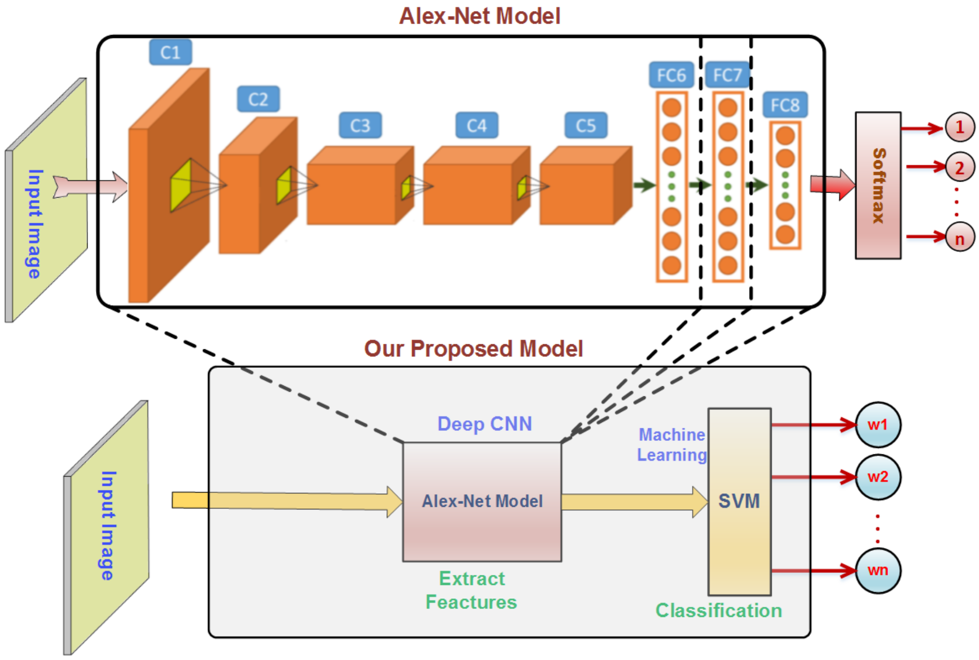



Figure 7.1 Proposed approach

## 7.2 Extracting Features and Classification using CNN

Writer identification techniques are used to identify a writer by processing a handwritten script to extract the discriminative features that can be used during the classification stage to decide to which writer the tested script belongs [12]. Although, a number of approaches have been proposed over the recent tears, the identification of handwritten scripts still seems to be a very challenging issue especially due to the fact that each writer may have a variety of handwriting styles that are subject to inter-writer and intra-writer variations [6].

Recently, CNNs have been studied and used as an effective approach for tasks such as image classification and object recognition. Typically, CNNs are trained on large datasets to extract intrinsic and discriminative patterns achieve high performance, but the long periods required for the training represents a big challenge with the use of such networks. For example, CNN has been used as a generic feature extractor and a previously trained network has been reused in order to manage this problem [135]. The technique uses the preserved learned neurons (kernels) so that the classification task is retrained based on different datasets. A similar strategy to overcome this issue was used by the authors in [121] for object recognition where the authors used a pre-trained CNN with pre-processed depth images. The features extracted by the CNN are then used by an SVM classifier for identifying the objects and the model was evaluated based on the Washington RGB-D Objects dataset.

Offline writer identification can be classified into two levels of analysis: texture-based and allograph-based. The latter approach depends on local descriptors extracted from tiny letter pieces (allographs). In this context, a method has been introduced for writer identification which uses the features extracted from CNNs as local descriptors [119]. Here, a global descriptor is created through a GMM supervector encoder and the results obtained were evaluated and compared against traditional local descriptors such as SIFT and SURF using the ICDAR 2013 and the CVL datasets showing attractive results.

Another interesting method is the model proposed by Feil et al. [122] where a CNN structure is employed for writer identification and retrieval. A feature vector is created for each writer which can then be compared with a dataset containing precalculated feature vectors. Their approach has been evaluated based on the ICDAR2013, ICDAR 2011 and CVL-Database datasets.

A similar approach to our work was presented by Mohsen et al. [136] for author identification tasks. The chapter proposed an approach using a DL method to extract the features for authorship identification. In the feature extraction phase, a stacked denoising auto-encoder (SDAE) was used with various parameters with the classification stage based on an SVM classifier. In addition, Elleuch et al. [137] presented an efficient model to integrate a CNN structure and SVM classifier for offline Arabic handwriting recognition where a dropout mechanism has been applied. Their proposed system swapped the trainable classifier of CNN with the SVM classifier and performance the was evaluated using HACDB and IFN/ENIT databases.

Recently, CNNs have been applied across different tasks and have achieved attractive results to solve many problems in different fields. One example is the work by Tatulli et al. in [138], who used a CNN architecture to recognise speech from images by extracting the features from ultrasound and video images produced by a video camera and an ultrasound imaging system.  Also, a DL technique has been used to present an effective method to identify norm conflicts in contracts [139].

### 7.3 Convolutional Neural Network Model

During recent years, major breakthroughs have been made in document recognition, object detection and many other tasks [135]. Deep CNN models have been designed based on the idea that the dimensions of images can be decreased by increasing the number of hidden layers consisting of convolutional and sampling layers. This allows for the extraction of discriminative features more efficiently from the lower dimensions. Due to their simple architecture and weight-sharing, CNNs involve fewer training parameters and neurons and it is then easier to perform the training process. Typically, a CNN is comprised of five different layers: input, convolution, pooling, fully-connected and output layers[130], [132]. The input layer specifies the size of the input image data. Then, in the next layer, the input patch is convolved based on different learned kernels using shared weights. Next, the pooling layer attempts to minimise the size of the image while preserving the information content. This process can be achieved by implementing a max-pooling process over the image parts of size 2x2 or 3x3 [132], [135]. These two layers are used for the feature extraction phase. Then, the extracted features are processed by allocating weight values which are then merged in the fully-connected layer. This step represents the classification phase of the convolutional network [119], [135]. However, to make the CNN work more efficiently and to provide significant results, it is essential to use large datasets at the training phase, but this comes at the expense of significant computation complexity. This is one of the main drawbacks of the model. Therefore, in this work, a well-known and commonly used model called AlexNet has been used with trained datasets in order to tackle this problem. The first part is used to extract the features, and classification is then performed using an SVM classifier instead of the softmax tool. In addition, a very efficient GPU, which is based on the NYIDIA GeForce GTX 770 with 16 GB of RAM, has been used in order to make training faster during the convolution operation. Although, a number of CNN architectures are available with including LeNet (1998), AlexNet (2012), ZF Net (2013), GoogLeNet (2014), VGGNet (2014) and, ResNet (2015) [124], AlexNet, which is also well-known deep CNN model [140], has been selected in this chapter. This system is a pre-trained model comprising of a number of hidden layers, including an input layer, five convolutional layers, three pooling layers, three fully-connected layers, and finally an output layer used to classify images into one thousand groups. Also, there are rectified linear units (ReLUs) after each layer. The main advantages of the model are that it is able to provide outstanding performance using fewer training parameters and with strong qualitative robustness.

However, the Alex-Net model requires the selection of convenient approaches in order to make the training process faster and also to avoid over-fitting caused by the complicated structure, the large numbers of training parameters and vast amounts of data. Consequently, the model has been constructed by inserting rectified linear unit (ReLU) nonlinearity in the data structures. In addition, the dropout approach has been employed into the fully connected layers FC6, FC7, and FC8 layers to enhance the robustness, which means that the learning process of the hidden layer is independent of the features extracted from the upper layer.

### 7.4 The proposed writer identification with Deep CNNs

In this chapter, the Alex-Net deep learning model [125] comprising of eight layers has been selected. The first five layers are convolutional layers while the last three are fully connected layers. In addition, there are other layers between the convolutional and connected layers: pooling and activation layers. A schematic representation of the network design and the steps of the feature extraction phase are illustrated in Figure (7.2).

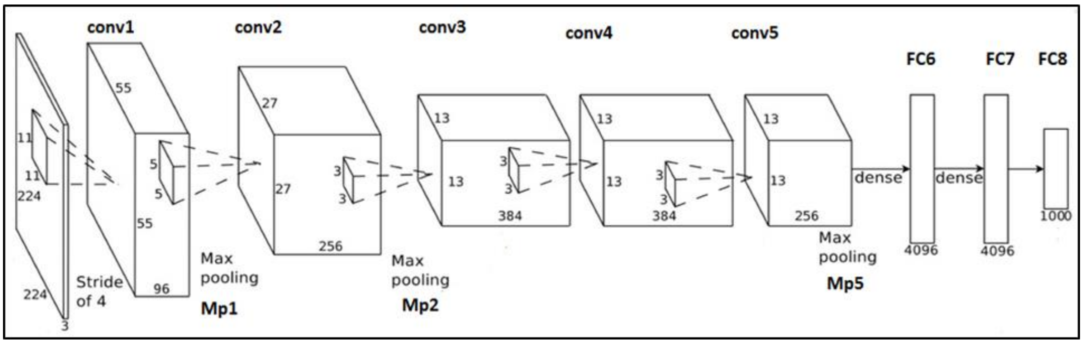



Figure 7.2 Flow chart of feature extraction based on Alex-Net Model

### 7.4.1 Input layer

The size of the input images is 227x227x3 with 'zero-centre' normalization procedure.

### 7.4.2 Convolutional Layer

As shown in Figure (7. 2), Conv1 is the first convolutional layer and represents the first step for feature extraction. Its output results in 96 feature maps of size 55x55. The feature maps have been created by using convolutional filters of size 11x11. In this layer, the input image of size 224x224 is processed using 96 convolutional kernels each of size 11x11 with a stride size of 4. At the end of this step, 96 feature maps are generated, each with a size of 55x55.

### 7.4.3 Pooling Layer

A pooling layer is one of the most important units of CNN. In this layer, the max-pooling algorithm has been employed to construct a two-dimensional pooling layer. As illustrated in Figure (7.2), Mp1 represents the first pooling layer and its aim is to reduce the spatial size of the feature maps in order to minimise the number of parameters and hence the complexity of the computational processes. In addition, this layer manages each feature map individually resulting in an output of 96 feature maps each having a size 27x27. These are generated by selecting the maximum value from the pooling domains. In this chapter, the max-pooling layer has been selected over a 3x3 area in order to control the speed of the dimensionality reduction task. The stride size was set to 2 resulting in an overlapped pooling strategy.

### 7.4.4 Convolutional Layer (Conv2)

The second convolution layer, Conv2, is also used to extract features and its specifications are very similar to the first convolution layer. The input data consisting of feature maps, received from Con1, are processed by using a set of 256 convolutional kernels each of size 5x5 and with stride, the size was 1 pixel. This allows us to extract 256 feature maps with a size of 27x27.

### 7.4.5 Other Convolutional and Pooling Layers

The remaining convolution layers Conv3, Con4, and Conv5 carry out similar tasks as the previous convolutional layers but some differences in terms of the values of the sizes and numbers of feature maps to extract further features.

### 7.4.6 Fully connected Layer (FC)

As shown above in Figure (7.2), the last three steps of the learning process are the fully connected layers, FC6, FC7 and FC8 which provide 4096, 4096 and 1000-dimensional feature vectors, respectively. Zero padding has been added during some phases of the learning process in order to create suitable sizes of feature maps. Moreover, the dropout technique[115], [128] has been utilised in the fully connected layers to reduce overfitting in the networks. The probability of the dropping ratio in the network is 0.5.

### 7.5 Data Sets

All experiments have been carried out on five datasets: three standard Arabic datasets, the Islamic Heritage project IHP [72], Qatar National Library QNL [73] and ICFHR-2012 [71], while the English datasets are Clusius [74] and IAM [68].

### 7.6 Experiments and Results

This section describes the performance evaluation of the proposed method using the datasets mentioned in the previous section. As explained above, the proposed approach as presented in Figure (7.2) is designed based on the Alex-Net model. It is composed of five convolutional layers, three fully- connected layers and the SVM classifier which uses the feature vectors extracted from FC6, FC7 or FC8 for classification. It is worth noting that an SVM classifier rather than Softmax because the SVM has been selected to improve the performance though at the expense of more computational complexity.

#### 7.6.1 Evaluation of Identification Performance

In our previous work, we presented a comparative study to investigate CNN model for writer identification method and its evaluation through its analysis against some of the traditional machine learning approaches, which means, the writer identification performances of machine learning algorithms using SURF and BoW approaches for feature extraction and SVM and KNN as classifiers have been compared against the performance of AlexNet model that has used fully connected layers FC6, FC7, and FC8 for features extraction and Softmax layer as classifier. Therefore, in this chapter, we are further investigating the reliability for writer identification with a view to further improve performance by examining and analysing the features extracted from the connected layers using SVM as a classifier with the AlexNet model. Therefore, the identification performance is evaluated using the features extracted from the fully connected FC6, FC7 or FC8 layers of the Alex-Net model. The dataset is divided into training and test data. The experiments have been carried out with datasets in two different scenarios. In the first one, the training data represents 70% of total images and the remainder, 30%, for the test data. While in the other scenario, 80% of data are used for the training and the remainder, 20%, is used for the testing. The partition process has been done in a random manner to avoid affecting the results. The number of images that have been processed from each dataset is mentioned above in section 5. In the first step, we extract the image features from one of the fully connected FC6, FC7 or FC8 layers in the Alex-Net model (one at each time). Then, these image features are used to train the SVM classifier in order to efficiently separate linearly two or more classes by using a linear transformation kernel called hyperplane. Moreover, since the system does not have any previous knowledge about the data, the SVM classifier is able to define the optimal hyperplane. In the last step, the test images can be classified by the trained SVM where the features are examined individually with respect to performance. The identification performance of our proposed approach based on the first scenario (70% - 30%) is shown in Table (7.1) and the results of the second scenario (80% - 20%) are shown in Table (7.2). In both cases, the results obtained indicate that the best performance in both cases is provided by FC7, with a little improvement in the second scenario.

| Dataset | FC6 (%) | FC7 (%) | FC8 (%) | Average(%) |
|---|---|---|---|---|
| Islamic Heritage Project (IHP) | 98.9 | 99.7 | 99.6 | 99.3 |
| Qatar National Library (QNL) | 95.0 | 99.0 | 98.7 | 97.5 |
| Clusius Dataset | 90.0 | 91.0 | 90.0 | 90.3 |
| IAM | 91.0 | 93.0 | 92.3 | 92.1 |
| ICFHR-2012 | 98.6 | 99.5 | 95.6 | 97.9 |

Table 7.1 Identification performance based on FC layers – the first scenario

Figure (7.3) below illustrates graphically the identification performance of the first scenario against each layer of the fully connected layers. It shows that image features extracted from FC7 provided the best identification performance among all other fully connected layers and for all selected datasets. As shown above, the features extracted from the FC7 layer yield the highest recognition accuracy while the features of layer FC8 show that the recognition accuracy decreases. This can be explained by the fact that layer FC8 of the Alex-Net extract information at a higher level the features extracted within layer FC7 layer have additional discriminative power useful in the classification task.

| Dataset | FC6 (%) | FC7 (%) | FC8 (%) | Average(%) |
|---|---|---|---|---|
| Islamic Heritage Project (IHP) | 98.9 | 99.7 | 99.6 | 99.3 |
| Qatar National Library (QNL) | 95.2 | 99.1 | 98.7 | 97.5 |
| Clusius Dataset | 90.3 | 91.5 | 90.6 | 90.3 |
| IAM | 91.5 | 94.0 | 93.0 | 92.1 |
| ICFHR-2012 | 98.6 | 99.5 | 96.3 | 97.9 |

Table 7.2  Identification performance based on FC layers – second scenario

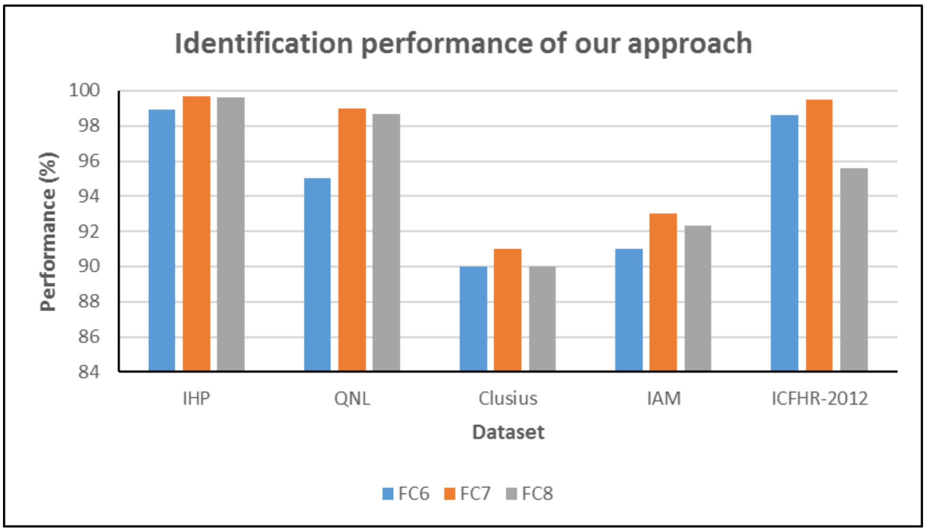



Figure 7.3 Identification performance of our proposed approach – first scenario

As shown above, in Table1, it can be clearly seen that the performance results obtained from all the selected data sets are relatively high, which proves that the features extracted by the deep CNN have even stronger discriminative ability and robustness. It can also be seen that the performance levels obtained based on the features from FC7 are more discriminative than those extracted by FC6 and FC8 layers. However, the difference between the results for the FC7 and FC8 is small for the QNL dataset, the performance is significantly improved from FC6 to FC7. Figure (7.4) below illustrates graphically the average identification performance for fully connected layers FC6, FC7 and FC8 evaluated using our dataset. The performance ranges from 90.3% to 99.3%, showing strong evidence that the proposed model is efficient and has achieved superior results compared to similar works as shown below in Table (7.2).

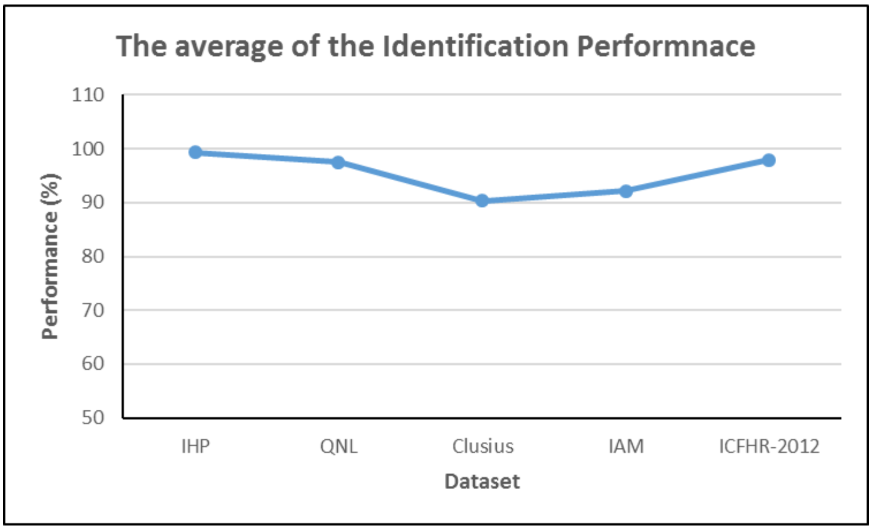



Figure7.4 Average performance for the dataset – first scenario

Furthermore, in this experiment, the results presented in Figure (7.3) are reshaped in Figure (7.5) for the purposes of a comparison of the layers and to show clearly that FC7 provides the best identification performance. The figure indicates clearly that the performance obtained using the ICFHR-2012 dataset outperforms the other datasets.

### 7.6.2 Comparison with Previous Works

In this section, the proposed approach is compared with some similar previous studies. As illustrated in Table 2, our proposed method clearly outperforms other listed works that have used the same operational conditions including the same datasets and the same number of subjects. The results show that our proposed model provides slightly improvements in identification performance compared to the previous state of the art approaches mentioned in Table (7.2). As mentioned previously, in this chapter we analyse the extracted learned features from a pre-trained CNN model, Alex-Net, followed by the SVM algorithm to perform classification. On the other hand, the extraction process of the features in the previous approaches [12], [69] mentioned in Table (7.3) is carried out using hand-crafted features such as OBI, grapheme, and

SURF. However, it is worth noting that improved performance achieved at the expense of more computational complexity since the feature extraction process requires much more time to complete.

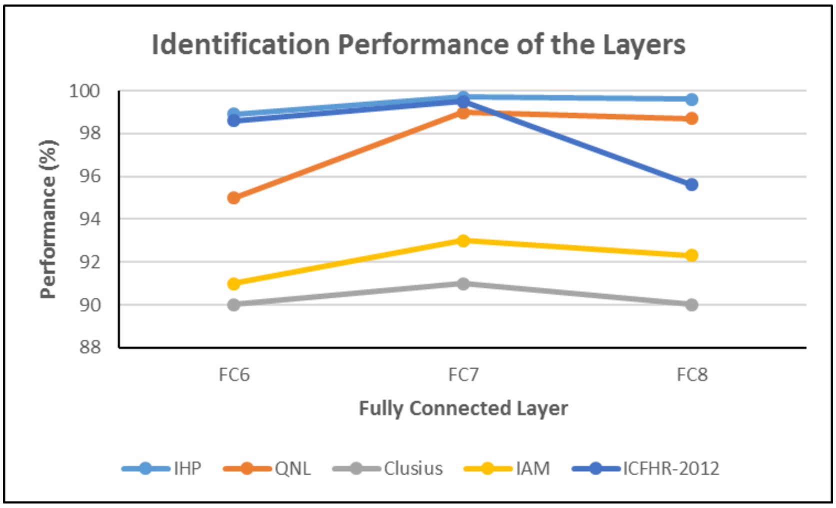



Figure 7.5 Comparison of the performance against each fully connected layer, the first scenario

| Approach developed by | Dataset | No. of writers | (Top1) Performance % |
|---|---|---|---|
| Authors [69] | IAM | 650 | 92 |
| Authors [69] | ICFHR2012 | 206 | 95 |
| Authors [12] | IAM | 650 | 92 |
| Authors [12] | ICFHR2012 | 228 | 97 |
| Our Approach | QNL | 121 | 99 |
| Our Approach | IHP | 29 | 99 |
| Our Approach | Clusius | 333 | 91 |
| Our Approach | IAM | 650 | 93 |
| Our Approach | ICFHR2012 | 228 | 99 |

Table 7.3 Performance comparison of our proposed approach with previous works.

### 7.7 Summary

In recent years, CNNs have emerged as a powerful technique for image recognition and computer vision. In this chapter, the Alex-Net model has been employed and the features extracted from the connected layers have been analysed to determine the discriminative power of the features. This has allowed us to feed the features to an

SVM to efficiently identify had writers. The selection and use of pre-trained Alexnet along with SVM provides the best performance compared to other algorithms, but some of the other approaches such as BoW can offer high accuracy. When our developed algorithms in chapter 6 (BoW) is compared against and our algorithm in this chapter based on the time consumed, the BoW method has shown to be more efficient than the neural networks. However, using CNN for classification requires more time for the training though the identification phase is not affected. The learned model has been evaluated based on different datasets including IHP, QNL, Clusius, IAM and ICFHR-2012 datasets. The results obtained suggest that the proposed model has achieved superior results and clearly shows that deep CNNs are more effective in extracting image features. It has also been proven that combining CNN features extracted from the connected layers can generate a more discriminative feature extractor. Finally, it can be clearly observed from the results that our proposed method provides a significant improvement in identification performance compared to state-of-the-art approaches in the identification of writers of handwritten text.

# Chapter 8

# Conclusion and Future Work

The main aim of this chapter is to summarise the main contributions regards to the research objectives and describe the general conclusions based on the findings of the studies presented in this work. After that, some suggestions are examined as future works of this research.

## 8.1 Summary of Thesis Contributions

Writer identification remains a challenging biometric recognition application. It is carried out as a pattern recognition problem to allocate an unknown written sample/pattern to one class (e.g., a writer) out of a set of classes (writers). Therefore, the process of writer identification can be defined as an algorithm/tool to assign a handwriting sample to one author/writer. While there are several writer identification systems have been developed for various applications including forensic science, document analysis, investigation of the historical documents, but this problem still receiving significant interest by the research community, because there are many issues are still unresolved such as insufficiency of datasets and handwriting material in different languages. The main aim of this work is to develop an accurate handwritten identification system by investigating new techniques and tools for the classification and analysis of ancient documents depending on different multi-scale features extraction techniques. In addition, another aim is to deploy the proposed techniques to support palaeographic studies by automatically deriving the writer of the relevant scripts. Therefore, the general objectives are to plan, analyse, design, build, and test novel classification algorithms and tools to support palaeographic analysis of historical Arabic manuscripts. In this work, some of automatic writer identification approaches using advanced machine learning techniques have been investigated and the obtained results led to the following contributions to knowledge:

- A novel algorithm for the segmentation and detection of text lines of grey-scale and colour historical documents has been proposed to evaluate medial seams and separating seams. The novelty relates to an efficient method to separate the two seams using a bilateral filter approach. The performance of our algorithm is dependent on the medial seam computation and separating seam computation using bilateral filtering. The experimental validation of the algorithm has been carried out using different datasets and the results obtained suggest that our proposed technique yields attractive results when compared against a few similar algorithms.
- Most existing handwriting identification systems use either statistical or model-based approaches. To further improve the performance, a novel writer identification approach has been presented using the concept of OBI feature extraction and its combination with the graphemes codebook method. The proposed algorithm has resulted in an improved identification performance when compared against similar techniques. In addition, to reduce the resulting high dimensionality of the feature vectors, a Kernel Principal Component Analysis (KPCA) has been used in our proposed method. Moreover, to further improve the identification performances, a number of nonlinear dimensionality reduction techniques have been investigated to get the optimum value of the system

performance such as Isomap, locally linear embedding (LLE), Hessian LLE and Laplacian Eigenmaps alongside KCPA have been used and evaluated. The obtained results show that the KPCA method has provided the best result compared to other reduction techniques used in this work. Also, the results proved that measuring and optimising the system performance of the writing identification based on dimensionality reduction techniques plays an important role in the writer recognition and handwriting classification.

- Further improvement of the identification performance can be achieved by using deep learning (convolutional neural networks) concept. Therefore, a comparative study of machine learning approaches for writer identification has been done, which means, the writer identification performances of machine learning algorithms using SURF and SIFT approaches for feature extraction and SVM and KNN classifiers have been compared against deep learning algorithms such as CNN (Alex-net model). The analysis was carried out using a comparative study using the IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting. These two datasets are widely used databases for writer identification. The obtained results show that deep learning outperforms machine learning algorithms.

- Finally, the Alex-Net model has been employed and the features extracted from the connected layers have been analysed to determine the discriminative power of the features. This has allowed us to feed the features to an SVM to efficiently identify had writers. The learned model has been evaluated based on different datasets including IHP, QNL, Clusius, IAM and ICFHR-2012 datasets. The results obtained suggest that the proposed model has achieved superior results and clearly shows that deep CNNs are more effective in extracting image features. It has also been proven that combining CNN features extracted from the connected layers can generate a more discriminative feature extractor. Finally, it can be clearly observed from the results that our proposed method provides a significant improvement in identification performance compared to state-of-the-art approaches in the identification of writers of handwritten text.

## 8.2 Future Work

In this research, our proposal as future work is to investigate and develop new techniques and tools for the classification and analysis of ancient documents to support palaeographic studies by automatically deriving the writer, date, and geographic location of the relevant scripts. Moreover, we should be able to identify the how many writers for each script. There are two main steps that have to achieve:

1. page layout analysis can be proposed to develop our work. The main idea is to try to separate the main body text from all the additional text and other content located on page margins in historical documents. using Gabor filter.
2. Apply a global refinement scheme, the refinement scheme is based on minimising an explicit energy function which is derived from properties of the text components, e.g., location, stroke width and area. We are planning to evaluate this approach on a larger dataset. But here we need to mention that the process of examining possible directions for segmenting side-notes regions according to their orientation is a demanding challenge as well.
3. We can try to investigate the efficiency of our proposed approach (presented in Chapter (4)) for other languages that may have different characteristics such as the Chinese language.

4. Apply the features that have been used in this work to recognise Arabic text in videos. We should try to locate Arabic text as the first step, then we can extract the target text from original video frames. After that, the text needs to be separated from the background.

.

In addition, our current datasets that we have used in our research did not offer the full range of problems or issues that we need to investigate and solve in complete document analysis and recognition system.

# References


[1] S. M and S. Mary Idicula, “A Survey on Writer Identification Schemes,” *Int. J. Comput. Appl.*, vol. 26, no. 2, pp. 23–33, 2011.

[2] S. M. Awaida, “State of the art in off-line writer identification of handwritten text and survey of writer identification of Arabic text,” *Educ. Res. Rev.*, vol. 7, no. 20, 2012.

[3] S. M. Awaida and S. A. Mahmoud, “Writer identification of arabic text using statistical and structural features,” *Cybern. Syst.*, vol. 44, no. 1, pp. 57–76, 2013.

[4] C. Djeddi, L. S. Meslati, I. Siddiqi, A. Ennaji, H. El Abed, and A. Gattal, “Evaluation of texture features for offline Arabic writer identification,” *Proc. - 11th IAPR Int. Work. Doc. Anal. Syst. DAS 2014*, no. April, pp. 106–110, 2014.

[5] S. M. Awaida, “State of the art in off-line writer identification of handwritten text and survey of writer identification of Arabic text,” *Educ. Res. Rev.*, vol. 7, no. 20, pp. 445–463, 2012.

[6] A. H. and K. S. Hicham EL MOUBTAHIJ1, “REVIEW OF FEATURE EXTRACTION TECHNIQUES FOR OFFLINE HANDWRITING ARABIC TEXT RECOGNITION,” *Int. J. Adv. Eng. Technol.*, vol. 5, no. 1 ©IJAET ISSN: 223119632012International Journal of Advances in Engineering & Technology, Mar. 2014. ©IJAET ISSN: 22311963, 2014.

[7] S. A. Mahmoud *et al.*, “KHATT: Arabic offline Handwritten Text Database,” *Proc. - Int. Work. Front. Handwrit. Recognition, IWFHR*, pp. 449–454, 2012.

[8] W. Yang, L. Jin, and M. Liu, “DeepWriterID: An End-to-End Online Text-Independent Writer Identification System,” *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 45–53, 2016.

[9] M. N. Abdi and M. Khemakhem, “Arabic writer identification and verification using template matching analysis of texture,” *Proc. - 2012 IEEE 12th Int. Conf. Comput. Inf. Technol. CIT 2012*, pp. 592–597, 2012.

[10] X. Wu, Y. Tang, and W. Bu, “Offline text-independent writer identification based on scale invariant feature transform,” *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 3,

pp. 526–536, 2014.

[11] A. Durou, A. Bouridane, and S. Almaadeed, “Text Line Extraction on Color and Grayscale Historical Manuscripts using Bilateral Filter,” in *3rd International Conference on Signal, Image, Vision and their Applications (SIVA’15) University of Guelma*, 2015, pp. 25–28.

[12] A. Durou, I. Aref, S. Al-Maadeed, A. Bouridane, and E. Benkhelifa, “Writer identification approach based on bag of words with OBI features,” *Inf. Process. Manag.*, no. August, pp. 1–14, 2017.

[13] A. Durou, I. Aref, M. Elbendak, S. Al-Maadeed, and A. Bouridane, “Measuring and optimising performance of an offline text writer identification system in terms of dimensionality reduction techniques,” *Proc. - 2017 7th Int. Conf. Emerg. Secur. Technol. EST 2017*, pp. 19–25, 2017.

[14] A. Durou, I. Aref, M. Elbendak, S. Al-Maadeed, and A. Bouridane, “A Comparative Study of Machine Learning Approaches for Writer Identification in Handwritten Documents,” pp. 1–9.

[15] K. Franke *et al.*, “WANDA: A generic Framework applied in Forensic Handwriting Analysis and Writer Identification.,” *Des. Appl. hybrid Intell. Syst. Proc. 3rd Int. Conf. Hybrid Intell. Syst.*, no. January, pp. 927–938, 2003.

[16] K. Franke and M. Köppen, “A computer-based system to support forensic studies on handwritten documents,” *Int. J. Doc. Anal. Recognit.*, vol. 3, no. 4, pp. 218–231, 2001.

[17] R. NIELS, L. VUURPIJL, and L. SCHOMAKER, “Automatic Allograph Matching in Forensic Writer Identification,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 21, no. 01, pp. 61–81, 2007.

[18] R. Plamondon and S. N. Srihari, “On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, 2000.

[19] B. Q. Huang, Y. B. Zhang, and M. T. Kechadi, “Preprocessing Techniques for Online Handwriting Recognition,” *Seventh Int. Conf. Intell. Syst. Des. Appl. (ISDA 2007)*, pp. 793–800, 2007.

[20] L. Schomaker and M. Bulacu, “Automatic writer identification using connected-component contours and edge-based features of uppercase western script,” *IEEE*

*Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 787–798, 2004.

[21] J. Hochberg, K. Bowers, M. Cannon, and K. Patrick, “Script and Language Identification for Handwritten Document Images,” *Int. J. Doc. Anal. Recognit.*, vol. 2, no. 2/3, pp. 45–52, 1999.

[22] G. Leedham and S. Chachra, “Writer identification using innovative binarised features of handwritten numerals,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2003-Janua, pp. 413–416, 2003.

[23] I. Bar-Yosef, I. Beckman, K. Kedem, and I. Dinstein, “Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents,” *Int. J. Doc. Anal. Recognit.*, vol. 9, no. 2–4, pp. 89–99, 2007.

[24] M. Bulacu and L. Schomaker, “Automatic handwriting identification on medieval documents,” *Proc. - 14th Int. Conf. Image Anal. Process. ICIAP 2007*, no. Iciap, pp. 279–284, 2007.

[25] M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy, “Automatic writer identification of ancient greek inscriptions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1404–1414, 2009.

[26] M. Bulacu and L. Schomaker, “Text-independent writer identification and verification using textural and allographic features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 701–717, 2007.

[27] a. Fornes, J. Llados, G. Sanchez, and H. Bunke, “Writer Identification in Old Handwritten Music Scores,” *2008 Eighth IAPR Int. Work. Doc. Anal. Syst.*, pp. 347–353, 2008.

[28] J. A. Rodríguez-Serrano, F. Perronnin, G. Sánchez, and J. Lladós, “Unsupervised writer adaptation of whole-word HMMs with application to word-spotting,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 742–749, 2010.

[29] T. Gilliam, “Writer identification in medieval and modern handwriting,” no. September, p. Writer identification in medieval and modern handw, 2011.

[30] A. Schlapbach and H. Bunke, “A writer identification and verification system using HMM based recognizers,” *Pattern Anal. Appl.*, vol. 10, no. 1, pp. 33–43, 2007.

[31] A. Bensefia and T. Paquet, “Writer verification based on a single handwriting word samples,” *Eurasip J. Image Video Process.*, vol. 2016, no. 1, 2016.

[32] A. Raj, A. Chaudhary, and A. professor, “A Review of Personal Identification Using Handwriting,” *Int. J. Eng. Sci. Comput.*, vol. 6, no. 4, pp. 3454–3457, 2016.

[33] A. Amin, “Off-line Arabic character recognition: The state of the art,” *Pattern Recognit.*, vol. 31, no. 5, pp. 517–530, 1998.

[34] A. Jailin Reshma, J. Jenushma James, M. Kavya, and M. Saravanan, “An overview of character recognition focused on offline handwriting,” *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 15, pp. 9372–9378, 2016.

[35] L. M. Lorigo and V. Govindaraju, “Offline Arabic Handwriting Recognition :,” vol. 28, no. 5, pp. 712–724, 2006.

[36] I. Siddiqi and N. Vincent, “Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features,” *Pattern Recognit.*, vol. 43, no. 11, pp. 3853–3865, 2010.

[37] B. Scholars, B. O. Access, and C. Commons, “University of Bradford eThesis Word Based Off-line Handwritten Arabic Classification and Recognition.”

[38] B. Al-Badr and R. M. Haralick, “Segmentation-free word recognition with application to Arabic,” *Proc. 3rd Int. Conf. Doc. Anal. Recognit.*, vol. 1, pp. 355–359, 1995.

[39] R. Plamondon and S. N. Srihari, “On-line and off-line handwriting recognition: A comprehensive survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, 2000.

[40] ∅Osivind Due Trier and A. K. Jain, “Goal-Directed Evaluation of Binarization Methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 12, pp. 1191–1201, 1995.

[41] R. Firdousi and S. Parveen, “Local Thresholding Techniques in Image Binarization,” *Int. J. Eng. Comput. Sci.*, vol. 3, no. 3, pp. 4062–4065, 2014.

[42] Y. Ying Liu and S. N. Srihari, “Document image binarization based on texture features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 540–544, 1997.

[43] D. Liu and J. Yu, “Otsu method and K-means,” *Proc. - 2009 9th Int. Conf. Hybrid Intell. Syst. HIS 2009*, vol. 1, no. 2, pp. 344–349, 2009.

[44] A. Namboodiri and A. K. Jain, “Document structure and layout analysis,” *Digit. Doc. Process.*, pp. 1–17, 2007.

[45] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, “Text line and word segmentation of handwritten documents,” *Pattern Recognit.*, vol. 42, no. 12, pp. 3169–3183, 2009.

[46] Z. Huang and P. Chan, “Content-based image retrieval using color moment and Gabor texture feature,” *Mach. Learn. …*, no. July, pp. 11–14, 2010.

[47] A. Amin, “Recognition of printed arabic text based on global features and decision tree learning techniques,” *Pattern Recognit.*, vol. 33, no. 8, pp. 1309–1323, 2000.

[48] G. Kim, V. Govindaraju, and S. N. Srihari, “An architecture for handwritten text recognition systems,” *Int. J. Doc. Anal. Recognit.*, vol. 2, no. 1, pp. 37–44, 1999.

[49] A. Elnagar and R. Bentrcia, “A Recognition-Based Approach to Segmenting Arabic Handwritten Text,” *J. Intell. Learn. Syst. Appl.*, no. November, pp. 93–103, 2015.

[50] T. Economy, “Chapter 4,” no. Oppenheim 1992, pp. 44–64, 2011.

[51] T. H. Reiss, “The revised fundamental theorem of moment invariants,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8. pp. 830–834, 1991.

[52] M. Stricker and A. Dimai, “Color indexing with weak spatial constraints,” *Storage Retr. Image Video Databases IV*, vol. 2670, pp. 29–40, 1996.

[53] V. S. Vyas and P. Rege, “Automated Texture Analysis with Gabor filter,” *Gvip*, vol. 6, no. 1, pp. 35–41, 2006.

[54] B S Manjunath and W Y Ma, “Texture features for browsing and retrieval of large image data,” vol. 18, no. 8, pp. 837–842, 1996.

[55] T. Arif, Z. Shaaban, L. Krekor, and S. Baba, “Object Classification Via Geometrical, Zernike and Legendre Moments,” *J. Theor. Appl. Inf. Technol.*, vol. 7, no. 1, pp. 31–37, 2009.

[56] D. Huang, C. Shan, M. Ardebilian, Y. Wang, and L. Chen, “Local Binary Patterns and Its Application to Facial Image Analysis : A Survey,” pp. 1–17.

[57] J. Ahmed, “Comparison of machine learning and deep learning classifiers for Traffic sign image classification.”

[58] E. Karami, S. Prasad, and M. Shehata, “Image Matching Using SIFT , SURF , BRIEF and ORB : Performance Comparison for Distorted Images Image Matching Using SIFT , SURF , BRIEF and ORB : Performance Comparison for Distorted Images,”

no. February 2016, 2015.

[59] R. Bala, “Survey on Texture Feature Extraction Methods,” *Int. J. Eng. Sci. Comput.*, vol. 7, no. 4, pp. 10375–10377, 2017.

[60] M. Zahedi, P. Dreuw, D. Rybach, T. Deselaers, and H. Ney, “Geometric Features for Improving Continuous Appearance-based Sign Language Recognition,” *Procedings Br. Mach. Vis. Conf. 2006*, pp. 104.1–104.10, 2006.

[61] and V. V. K. B.V. Ramana Reddy1, A. Suresh2, M. Radhika Mani3, “Classification of Textures Based on Features Extracted from Preprocessing Images on Random Windows,” *Int. J. Adv. Sci. Technol. Vol. 9, August, 2009*, vol. 1, no. 1, pp. 15–18, 2009.

[62] H. Bhavsar and A. Ganatra, “Support Vector Machine Classification using Mahalanobis Distance Function,” *Int. J. Sci. Eng. Res.*, vol. 6, no. 1, pp. 618–626, 2015.

[63] A. A. Hasseim, R. Sudirman, and P. I. Khalid, “Handwriting Classification Based on Support Vector Machine with Cross Validation,” *Engineering*, vol. 5, no. May, pp. 84–87, 2013.

[64] M. Gopinathan and P.-C. Berg, “A Deep Learning Ensemble Approach to Gender Identification of Tweet Authors,” no. June, 2017.

[65] P. Barrett, “Euclidean Distance Whitepaper,” *Tech. Whitepaper Ser. 6*, p. 26, 2005.

[66] R. G. Brereton, “The chi squared and multinormal distributions,” *J. Chemom.*, vol. 29, no. 1, pp. 9–12, 2015.

[67] IEEE UAE Student Branch, “Distance in Classification,” 2016.

[68] U. V. Marti and H. Bunke, “The IAM-database: An English sentence database for offline handwriting recognition,” *Int. J. Doc. Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, 2003.

[69] E. Khalifa, S. Al-Maadeed, M. A. Tahir, A. Bouridane, and A. Jamshed, “Off-line writer identification using an ensemble of grapheme codebook features,” *Pattern Recognit. Lett.*, vol. 59, pp. 18–25, 2015.

[70] M. Bulacu, L. Schomaker, and A. Brink, “Text-independent writer identification and verification on offline arabic handwriting,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2, pp. 769–773, 2007.

[71] A. Hassaïne and S. Al Maadeed, "ICFHR2012 competition on writer identification - Challenge 2: Arabic scripts," *Proc. - Int. Work. Front. Handwrit. Recognition, IWFHR*, pp. 835–840, 2012.

[72] Http://ocp.hul.harvard.edu/ihp/, "Harvard University, Islamic Heritage Project (IHP)." [Online]. Available: http://ocp.hul.harvard.edu/ihp/.

[73] Q. D. Library, "Articles From Our ExpertsNo Title." [Online]. Available: , https://www.qdl.qa/en/articles-from-our-experts?themes[0]=634.

[74] Http://www.dwc.knaw.nl/biografie/clusius/digital-edition-of-the-clusius-correspondence/., "For the life and work of Clusius and his many correspondents, see esp. Egmond 2010.," 2010. .

[75] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, p. 10, 2007.

[76] C. Tomasi, "Bilateral Filtering for Gray and Color Images," 1998.

[77] N. Arvanitopoulos and S. Susstrunk, "Seam Carving for Text Line Extraction on Color and Grayscale Historical Manuscripts," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2014-Decem, no. Ic, pp. 726–731, 2014.

[78] R. Saabni, A. Asi, and J. El-Sana, "Text Line Extraction for Historical Document Images," 2014.

[79] N. Ouwayed and A. Belaïd, "A general approach for multi-oriented text line extraction of handwritten documents," pp. 297–314, 2012.

[80] J. Kumar, W. Abd-Almageed, L. Kang, and D. Doermann, "Handwritten Arabic text line segmentation using affinity propagation," *Proc. 8th IAPR Int. Work. Doc. Anal. Syst. - DAS '10*, pp. 135–142, 2010.

[81] F. Lebourgeois, H. Emptoz, E. Trinh, and J. Duong, "Networking Digital-Document Images," pp. 1–5, 2001.

[82] C. Clausner, A. Antonacopoulos, and S. Pletschacher, "A robust hybrid approach for text line segmentation," *21st Int. Conf. pattern Recognit.*, no. Icpr, pp. 335–338, 2012.

[83] M. Liwicki, E. Indermuhle, and H. Bunke, "On-line handwritten text line detection using dynamic programming," *Doc. Anal. …*, vol. 1, no. Icdar, 2007.

[84] R. Kasturi, L. O'Gorman, and V. Govindaraju, "Document image analysis: A

primer," *Sadhana*, vol. 27, no. 1, pp. 3–22, 2002.

[85] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, "Binarization-free text line segmentation for historical documents based on interest point clustering," *Proc. - 10th IAPR Int. Work. Doc. Anal. Syst. DAS 2012*, pp. 95–99, 2012.

[86] G. Nagy, S. Seth, and M. Viswanathan, "A Prototype Document Image Analysis System for Technical Journals," *Computer (Long. Beach. Calif).*, vol. 25, no. 7, pp. 10–22, 1992.

[87] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Sixth Int. Conf. Comput. Vis. (IEEE Cat. No.98CH36271)*, pp. 839–846, 1998.

[88] S. Srisuk, "Bilateral filtering as a tool for image smoothing with edge preserving properties," *2014 Int. Electr. Eng. Congr. iEECON 2014*, no. x, pp. 0–3, 2014.

[89] "Juma Al-Majid Center for Culture and Heritage, online, accessed June 2011. <http:// www.almajidcenter.org>." .

[90] "Wadod center for manuscripts, online; accessed June, 2011. <http://wadod.com>." .

[91] "American University of Beirut,http://ddc.aub.edu.lb/," *2011*. .

[92] M. Bulacu, L. Schomaker, and L. Vuurpijl, "Writer identification using edge-based directional features," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2003-Janua, pp. 937–941, 2003.

[93] M. Kam, D. Ph, G. Fielding, M. Sc, and R. Conn, "Writer Identification by Professional Document Examiners † Moshe Kam, * Ph.D.; Gabriel Fielding, * M.Sc.; and Robert Conn, ** Ph.D.," pp. 1–33.

[94] D. Paraskevas, G. Stefanos, and K. Ergina, "Writer Identification Using a Statistical and Model Based Approach," *2014 14th Int. Conf. Front. Handwrit. Recognit.*, pp. 589–594, 2014.

[95] A. J. Newell and L. D. Griffin, "Writer identification using oriented Basic Image Features and the Delta encoding," *Pattern Recognit.*, vol. 47, no. 6, pp. 2255–2265, 2014.

[96] Y. Hannad, I. Siddiqi, and M. E. Y. El Kettani, "Writer identification using texture descriptors of handwritten fragments," *Expert Syst. Appl.*, vol. 47, pp. 14–22, 2016.

[97] S. Fiel and R. Sablatnig, "Writer Retrieval and Writer Identification Using Local Features," *2012 10th IAPR Int. Work. Doc. Anal. Syst.*, pp. 145–149, 2012.

[98] Z. A. Daniels and H. S. Baird, “Discriminating features for writer identification,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, pp. 1385–1389, 2013.

[99] J. L. Vasquez, C. M. Travieso, and J. B. Alonso, “Off line writer identification based on graphometric parameters,” *INES 2013 - IEEE 17th Int. Conf. Intell. Eng. Syst. Proc.*, no. Mv, pp. 219–223, 2013.

[100] G. Ghiasi and R. Safabakhsh, “Offline text-independent writer identification using codebook and efficient code extraction methods,” *Image Vis. Comput.*, vol. 31, no. 5, pp. 379–391, 2013.

[101] M. N. Abdi and M. Khemakhem, “A model-based approach to offline text-independent Arabic writer identification and verification,” *Pattern Recognit.*, vol. 48, no. 5, pp. 1890–1903, 2015.

[102] L. D. Griffin, M. Lillholm, M. Crosier, and J. Van Sande, “Basic image features (BIFs) arising from approximate symmetry type,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5567 LNCS, pp. 343–355, 2009.

[103] M. Crosier and L. D. Griffin, “Texture Classification with a Dictionary of Basic Image Features,” 2008.

[104] A. J. Newell and L. D. Griffin, “Natural image character recognition using oriented basic image features,” *Proc. - 2011 Int. Conf. Digit. Image Comput. Tech. Appl. DICTA 2011*, no. May, pp. 191–196, 2011.

[105] L. Schomaker, K. Franke, and M. Bulacu, “Using codebooks of fragmented connected-component contours in forensic and historic writer identification,” *Pattern Recognit. Lett.*, vol. 28, no. 6, pp. 719–727, 2007.

[106] M. Bulacu and L. Schomaker, “Combining Multiple Features for Text-Independent Writer Identification and Verification,” *Proc. 10th IWFHR*, pp. 281–286, 2006.

[107] K. Arunasakthi and L. KamatchiPriya, “A Review on linear and non-linear dimensionality reduction,” *Mach. Learn. Appl.*, vol. 1, no. 1, pp. 65–76, 2014.

[108] J. P. Cunningham and Z. Ghahramani, “Linear Dimensionality Reduction: Survey, Insights, and Generalizations,” vol. 16, pp. 2859–2900, 2014.

[109] I. Ross, “Nonlinear Dimensionality Reduction Methods in Climate Data Analysis,” no. September, 2009.

[110] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for non linear dimensionality reduction,” *Science (80-. ).*, vol. 290, no. December, pp. 2319–2323, 2000.

[111] B. Schölkopf, A. Smola, and K. R. Müller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[112] S. Pr, “GPU Assisted Implementation of Kernel-Based Template Attacks,” 2011.

[113] P. Aljabar, R. Wolz, and D. Rueckert, “Manifold Learning for Medical Image Registration, Segmentation, and Classification,” pp. 351–372.

[114] M. T. for D. R. L.J.P. van der Maaten, *https://lvdmaaten.github.io/drtoolbox*. .

[115] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[116] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn, “Deep Feature Learning for EEG Recordings,” 2015.

[117] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*, vol. 521, no. 7553. 2017.

[118] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” *AISTATS ’11 Proc. 14th Int. Conf. Artif. Intell. Stat.*, vol. 15, pp. 315–323, 2011.

[119] V. Christlein, D. Bernecker, A. Maier, and E. Angelopoulou, “Offline Writer Identification Using Convolutional Neural Network Activation Features,” *Proc. Ger. Conf. Pattern Recognit.*, vol. 9358, pp. 540–552, 2015.

[120] S. Loussaief and A. Abdelkrim, “Machine learning framework for image classification,” *2016 7th Int. Conf. Sci. Electron. Technol. Inf. Telecommun.*, pp. 58–61, 2016.

[121] M. Schwarz, H. Schulz, and S. Behnke, “RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features,” *2015 IEEE Int. Conf. Robot. Autom.*, no. May, pp. 1329–1335, 2015.

[122] S. Fiel and R. Sablatnig, “Writer Identification and Retrieval using a Convolutional Neural Network,” *Proc. Int. Conf. Comput. Anal. Images Patterns*, pp. 26–37, 2015.

[123] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12-June, pp. 3431–3440, 2015.

[124] C. Paper, “Application of Deep Learning to Computer Vision : A Comprehensive Study Application of Deep Learning to Computer Vision : A Comprehensive Study,” *2016 5th Int. Conf. Informatics, Electron. Vis.*, vol. 2016 5th I, no. May, pp. 592–597, 2016.

[125] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[126] H. Liu, H. Tang, W. Xiao, Z. Guo, L. Tian, and Y. Gao, “Sequential Bag-of-Words model for human action classification,” *CAAI Trans. Intell. Technol.*, vol. 1, no. 2, pp. 125–136, 2016.

[127] X. Niu, “Fusions of Cnn and Svm Classifiers for Recognizing Handwritten Characters,” no. April, 2011.

[128] D. R. Tobergte and S. Curtis, “Improving Neural Networks with Dropout,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[129] K. L. and W. Z. X. Li, G. Zhang, “Deep Learning and its Parallelization,” Chapter In book: Big Data, DOI: 10.1016/B978-0-12-805394-2.00004-0, 2016, pp. , pp.95–118.

[130] Z. W. Rawat Waseem, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” *Neural Comput. 29(9)1-98 · June 2017 with*, no. October, 2017.

[131] S. Yu, S. Jia, and C. Xu, “Convolutional neural networks for hyperspectral image classification,” *Neurocomputing*, vol. 219, pp. 88–98, 2017.

[132] L. Tobias, A. Ducournau, F. Rousseau, G. Mercier, and R. Fablet, “Convolutional Neural Networks for object recognition on mobile devices: A case study,” *2016 23rd Int. Conf. Pattern Recognit.*, pp. 3530–3535, 2016.

[133] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, “Face Recognition Using Convolutional Neural Network and Simple Logistic Classifier,” *Soft Comput. Ind. Appl.*, pp. 197–207, 2014.

[134] B. A. Durou Amal, Aref Ibrahim, Elbendak Mosa, Al-Maadeed Somaya, “A Comparative Study of Machine Learning Approaches for Writer Identification in Handwritten Documents,” *Pap. has been Submitt. to EUVIP2018 Wait. Notif.*

[135] L. Hertel, E. Barth, T. Kaster, and T. Martinetz, “Deep convolutional neural networks as generic feature extractors,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2015-Septe,

2015.

[136] A. M. Mohsen, “Author Identification using Deep Learning,” *Proc. - 2016 15th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2016*, pp. 898–903, 2016.

[137] V. Christlein, D. Bernecker, F. Hönig, and E. Angelopoulou, “Writer Identification and Verification Using GMM Supervectors,” *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pp. 998–1005, 2014.

[138] T. H. Eric Tatulli, “Feature Extraction Using Multimodal Convolutional Neural Networks for Visual Speech Recognition,” pp. 2–6, 2017.

[139] J. P. Aires and F. Meneguzzi, “Norm Conflict Identification using Deep Learning.”

[140] A. Ibrahim, A. L. Abbott, M. E. Hussein, and C. V May, “Input Fast-Forwarding for Better Deep Learning.,” pp. 1–8.