ABSTRACT

The researches in Optical Character Recognition (OCR) area by using Hidden Markov Models (HMMs) are continuing until this moment.

The work presented in this thesis is proposed for recognition of offline isolated Arabic handwritten characters using HMMs as classifier and Freeman chain code as feature extraction method. We use scheme to recognize the bodies of the characters only since many characters might share the same body. Characters with similar bodies are grouped as one class, then each class represented by one-character body. This scheme groups 34 characters by 18 character bodies. The mean reason behind this idea is to overcome the character segmentation problem

This work is divided into three main phases to provide a complete recognition system. The first phase is the preprocessing, which applies efficient preprocessing methods which are essential for optical character recognition. In this phase, methods for normalization and digitization are implemented. Then dots are removed from the characters, then characters' bodies are thinned. The second phase is feature extraction. This phase makes use of the thinned images to extract features that are essential in recognizing the images. Features are extracted by implementing Freeman chain code (FCC) method, and then it normalized to 10 digits for each sample. The third and final phase is the classification of characters' bodies by Hidden Markov Models (HMMs) classifier.

One of the important finding of this thesis is the identification of high confusion between some classes, this due to the variation of writers' styles which case similarity between characters. Moreover, error may occur due to inadequate capability with the features used.

The proposed system has been implemented and tested on MATLAB R2010b environment. The dataset used in this thesis is Isolated Handwritten Arabic Characters (IHAC) dataset, which collected by Arabic Language Technology Research Group at Sudan University of Science and Technology.

المستخلص

مازالت البحوث مستمرة في مجال هذا البحث وذلك باستخدام نماذج ماركوف المخفية حتى يومنا هذا. تعمل الدراسة في التعرف على حروف اللغة العربية غير المتصلة المكتوبة يدوياً باستخدام نماذج ماركوف المخفية للتصنيف واستخلاص الخواص باستخدام سلاسل فريمان.

في هذه الأطروحة تم للتعرف على اجسام الحروف، بما أن معظم الحروف لها اجسام مشابهة، تم استخدام نهج لتجميع الحروف المتشابهة الأجسام في قسم واحد وكل قسم تم تمثيله بحرف واحد. بهذه الطريقة عدد الحروف تناقص من ٣٤ حرف الي ١٨ حرف. الهدف من هذه الفكرة هو التخلص من عملية تجزئة الحرف لما بها من صعوبات.

تم تقسيم هذا العمل إلى ثلاثة مراحل اساسية وذلك للتعرف على النظام. المرحلة الاول وهي التجهيز، والتي تقوم على تطبيق طرق اساسية للتعرف على الحروف. في هذه المرحلة يتم تطبيق أحرف خاصه للتطبيع والترقيم. ومن ثم ازالت النقاط من بعض الحروف وتنحيف أجسام الحروف. المرحلة الثانية استخلاص الخواص. هذه المرحلة تقوم باستخدام الحروف ذات الصور غير الواضحة أو المبهمة لاستخلاص السمات أو خواص اساسية لإدراك هذه الصور.

ثم استخلاص الخواص أو الميزات وذلك عن طريق تطبيق خوارزمية فريمان ومن ثم تطبيع هذه الخواص الى عشرة ارقام لكل نموذج. المرحلة الثالثة والاخيرة هي عبارة عن التعرف على الحروف باستخدام نماذج ماركوف المخفية (HMMs).

من اهم مساهمات في هذه الدراسة تحيد التشابه العالي في بعض مجموعات الحروف ، هذا بسبب الاختلاف في طريقة الكتابة للكاتبين مما يؤدي الي التشابه بين الحروف بالإضافة الي ذلك الأخطاء قد تحدث بسبب عدم دقة الخواص المستخدمة.

البيانات المستخدمة في هذه الأطروحة هي عبارة عن حروف اللغة العربية المكتوبة بخط اليد التي تم الحصول عليها بواسطة مجموعة بحث تقنية اللغة العربية في جامعة السودان للعلوم والتكنلوجيا. كما تم تطبيق هذا النظام المقترح على بيئة الماتلاب R2010b.

DEDICATION

I am dedicating this effort to my father's soul, my beloved mother for her sincere prayers and support, my supervisor for always being supportive and encouraging, my lovely husband who gave me great support, my brothers and sisters for their continuous support, my kids who suffered more with me, my friends, my colleagues, and to all Muslim Umma.

ACKNOWLEDGMENT

“Praise be to Allah, the cherisher and the sustainer of the world”, “praise be to him he who taught by the pen, taught man, that which he did not know” First, my strongest thanks to ALLAH " الله" the most merciful, without his help and blessing, this PHD thesis would not complete.

My sincere thanks are expressed to my supervisor, Dr. Mohammed ElHafiz Mustafa Musa for all his guidance, support, bright ideas, continuous encouragement, integral view on thesis, and for his assistance in the preparation of this document. Many thanks to Dr. Talaat Wahby for his great support and help. Special thanks to my mother for always being there for me and continually providing me with love and encouragement. I would like to express my grateful gratitude to my lovely husband, Mohamed Fathelrahman for all his support during the long period that the thesis took up. My special thanks to my sisters for their loving support and continuous help. I would like to thank my brothers for their continuously supportive phone calls. Great thanks are extended to Dr. Najda Mohamed Abdelrahim for her continuous encouragement. My thanks extended to my friends and colleagues who provided me with emotional support. Finally, I would like to thank all staff members of Sudan University of Science and Technology Faculty of Computer Science and Information Technology for their emotional support.

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
AI	Artificial Intelligence
BPNN	Back Propagation Neural Network
DCT	Discrete Cosine Transform
DHMM	Discrete Hidden Markov Models
DLDA	Diagonal Linear Discriminant Analysis
DQDA	Diagonal Quadratic Discriminant Analysis
EM	Expectation Maximization
FCC	Freeman Chain codes
GHMM	Gaussian Hidden Markov Models
GT	Ground Truth information
HCR	Handwritten Character Recognition
HMMs	Hidden Markov Models
IFHCDB	Isolated Farsi Handwritten Character Data Base
IFN/ENIT	Institute of Communications Technology/ Ecole Nationale d'Ing'enieurs de Tunis
KB	Kilo Byte
KLT	Karhunen-Loève Transform
kNN	K Nearest Neighbor
LDA	Linear Discriminant Analysis
LM	Local Minima
LMS	Least Mean Square
MGHMM	Mixture of Gaussian Hidden Markov Models
ML	Maximum Likelihood
MLP	Multi-Layer Perceptron
NN	Neural Network

OCR	Optical Character Recognition
PR	Pattern Recognition
QDA	Quadratic Discriminant Analysis
RBF	Radial Basis Function
SUST/ ALT	Sudan University of Science and Technology / Arabic Language Technology group
SVM	Support Vector Machines
VA	Viterbi Algorithm
VQ	Vector Quantization
VSP	Valid Segmentation Points

CHAPTER ONE

INTRODUCTION

1.1 Overview

Electronic document management systems provide great benefits to society. Software tools such as word processors are used in the generation, storage and retrieval of documents in a variety of formats. Using such tools, electronic documents

can be edited, printed, or distributed across networks[1]. However, handwritten text cannot be manipulated by computer, so there is a need to extract the information in such documents to store them in a computerized format. The solution for this task is in the branch of pattern recognition known as optical character recognition (OCR).

In the recent years optical character recognition systems has over a lot of attention in the field of pattern recognition, because it is applicable in various fields. It aims to convert all the manual document in digital environment[2]. In other words, it converts the handwritten characters from the image form into machine editable form. Handwritten character recognition systems can improve the interaction between human and machine, which facilitates the application. The recognition of handwritten characters can be very useful, when large volume of handwritten characters need to be processed. On the other hand, building a recognition system for handwritten character seems to be a difficult task[3].

Since optical character recognition has great benefits to human and after carefully studying the problems for recognized isolated Arabic characters, we attempt to construct recognition classifier. This classifier design by using hidden Markov models (HMMs).

1.2 Problem Definition

Isolated handwritten characters, especially Arabic characters, usually contains more than one part, which causes difficulty to decide which part to be recognized first. Recognition of such characters is important in office application as well as in many other applications. Therefore, this thesis aims to build offline recognition system for isolated Arabic handwritten characters using Hidden Markov Models (HMMs).

Due to the distinctive nature of Arabic characters, some characters have the same shape but differ in the number of dots and position as will be described in chapter two. Therefore, we use scheme to recognize the bodies of the characters only since many characters might share the same body. Characters with similar bodies are classified as one class, then each class represented by one-character body. This scheme decreases the number of the characters in dataset from 34 characters to 18 characters. Moreover, the processing speed will be increased, and the media storage will be decreased.

Using the Freeman Chain Code (FCC) to extract character body (shape) features seems to be most appropriate, because it works well in connected object. This thesis proposed two techniques to generate and normalize FCC.

The system based on holistic approach, which is more suitable in character recognition, the characters recognized as whole without segmentation.

There are four problems considered to drive this thesis, which can be summarized as follows:

- The thesis studies an offline recognition system, which differs from the online recognition system.
- The thesis deals with Arabic language, which differs from other languages in many aspects.

- The thesis studies isolated characters, which differs from other character form. It's an important one, so an isolated character is the priority in this thesis.
- The thesis uses FCC to extract features, then pass it to HMM for classification purpose.

1.3 Thesis Objectives

The main objective of this thesis is the automatic reading of offline isolated Arabic handwritten characters' bodies by developing an Arabic OCR system to recognize the isolated characters' bodies' efficiency. To achieve the main objective of this thesis the following sub-objectives are addressed:

- Building pre-processing phase for recognizing offline Arabic handwritten characters.
- Developing feature extraction phase for recognizing offline handwritten characters. Which implemented by extracted and normalized the freeman chain code of the character's bodies.
- Constructing a simple classifier for the character body by using HMMs.
- Building a suitable code.

1.4 Contribution

The contribution of this thesis can be summarized as:

1. Developing an OCR system for recognition of isolated Arabic handwritten characters' bodies. The proposed system is expected to provide new prospective of characters recognition by grouping similar character's bodies into one class, then each class represented by one character body, and accordingly storage will be maintained and retrieval time will be reduced.

2. A second contribution is in the field of the dataset, SUST\ALT isolated dataset is a very large dataset contains 28 Arabic characters in addition to 6 supplementary characters. It consists of more than 47,000 samples. All these characters used to train and test the proposed system.
3. A new framework based on the chain code representation is proposed. In this framework, we combined chain code as feature extraction technique and HMM as classifier.

1.5 Thesis Organization

The thesis included six chapters; chapter one presented the introduction of the thesis, it summarizes the thesis problem, the objective, thesis contribution and followed by thesis organization.

Chapter two described the concept of OCR, and a brief review and background about OCR. Furthermore, it discusses the main characteristics of Arabic language and its difficulties from other languages. Also, the chapter discusses the phases involved in the OCR system. Which are summarized as: data capture, preprocessing, feature extraction, classification, and post processing.

It also surveys previous works in offline handwritten. Since this research study an isolated Arabic Handwritten characters, a survey of systems in this area is presented.

Because HMM is used as main classifier in this thesis, chapter three discussed all issues related to this method.

First the theoretical background of HMM is introduced. Then the three problems of HMM and their solutions are discussing. Moreover, the commonly Arabic handwritten datasets are described. Furthermore, the three topologies of HMM are discussed. Finally, the applications of HMM in: speech, text recognition, phoneme recognition, offline signature, protein domain are discussed.

Chapter four is about the research methodology, it explains the complete implementation of handwritten recognition system methods.

Chapter five presents the experimental result and discussion.

Finally, chapter six summarizes the conclusions and future works.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

In this chapter, the concepts of OCR approaches are described in (Section 2.2). Then the main characteristics of Arabic language are discussed in (Section 2.3). Then phases involved in OCR system are discussed as general in (Section 2.4), these phases are: preprocessing, segmentation, feature extraction and classification.

The main approaches for text recognition are described in (Section 2.5). Many studies need to found ground trust information about the freely dataset. Therefore, in (Section 2.6) we attempted to discuss some commonly used datasets, e.g. SUST/ALT, IFN/ENIT, and IFHCDB.

The objective of the current chapter is to summarize different works accomplished in offline Arabic recognition systems.

Special attention is given to isolated handwritten Arabic character recognition systems, since this thesis focuses on this subject. Therefore, a quick survey of work done on this field is introduce in (Section 2.7).

2.2 The Optical Character Recognition Approaches (OCR)

Pattern recognition (PR) has a long history, but before the 1960s it was mostly the output of theoretical research in statistics. As with everything else, the advent of computers increased the demand for practical applications of pattern recognition, which in turn set new demands for further theoretical developments[4].

The Optical Character Recognition (OCR) is one of important research area in pattern recognition[5].

It has many definitions, OCR defined as a process that attempts to turn a paper document into a fully editable form, which can be used in word-

processing and other applications as if it had been typed through the keyboard[6].

Also OCR was defined by Srihari et al.as the task of transforming text represented in the special form of graphical marks into its symbolic representation[7] .

The recognition of handwritten can be applied in many areas such as names of persons, companies, organizations, newspapers, letters, archiving and retrieving texts, proteins and genes in the molecular biology context, journals, books, bank chequs, personal signatures and digital recognition, etc.[8],[9].

A recognition system can be either online or offline[10]. It is online if the data being captured during the writing process. It always captured by special pen on an electronic interface.

Online recognition has several interesting characteristics: firstly, recognition is performed on one dimensional rather than two dimensional images, secondly, the writing line is represented by a sequence of dots which its location is a function of time[8].

A recognition system is offline, if its data scanned by scanner after writing process is over[11], such as any images scanned in by a scanner. In this case, only the image of the handwriting is available.

When we compared online handwriting recognition systems with offline systems, we found that offline systems are considered more difficult than online systems. This difficulty due to several reasons, out of which online handwriting recognition depends on temporal information[12], which facilitate the recognition system, but the temporal information are lacked in offline handwriting, it depends on passive images stored in files. This lemma makes offline systems less accurate than online systems.

Furthermore, offline systems are more complex than online systems, because they depend on human writing which had more feature and characteristic specially for Arabic language [13]. Summarizes the differences between online and offline recognition systems.

Table (2.1): The Differences between Online and Offline Recognition Systems:

Criteria of Recognition	Online Recognition System	Offline Recognition System
Data Capture	During the writing process	Scanned in by a scanner or camera.
Data Type	Temporal information	Not temporal information
Accuracy	More accurate	Less accurate
Complicity	Less complex	More complex

2.3 General Characteristics of Arabic Language

Many studies have been conducted on recognition of Chinese, Japanese and Latin languages, but few were done on Arabic handwritten recognition[14]. One of the main reasons for this is that characteristics of Arabic language do not allow direct implementation of many algorithms used in other languages.

The characteristics of Arabic language can be summarized as follows:

Arabic language is represented in 28 characters and appears in different four shapes isolated, initial, medium or final, these different shapes were listed in Table (2.2).

Arabic language is written from right to left, rather than from left to right this is useful for human reader rather than for the computer see Figure (2.1)

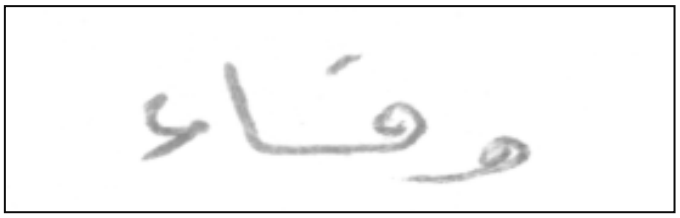


Figure (2.1): Arabic Word Written from Right to Left

Some Arabic character have the same shape and differ in the number of dots by which it will be identified, for example characters ب، ت، ث have the same shape but differ in number of dots, one dot in character Baa, two dots in character Taa, and three dots in character Thaa.

Table (2.2):The Different Shapes of Arabic Characters[15]

Serial No	Character	Isolated	Initial	Medium	Final
1	Alef	أ	ا	ـا	ا
2	Baa	ب	بـ	ـبـ	ـب
3	Taa	ت	تـ	ـتـ	ـت
4	Thaa	ث	ثـ	ـثـ	ـث
5	Jeem	ج	جـ	ـجـ	ـج
6	Haa	ح	حـ	ـحـ	ـح
7	Khaa	خ	خـ	ـخـ	ـخ
8	Dal	د	د	ـد ، د	ـد ، د
9	Thal	ذ	ذ	ـذ ، ذ	ـذ ، ذ
10	Raa	ر	ر	ـر ، ر	ـر ، ر
11	Zay	ز	ز	ـز ، ز	ـز ، ز
12	Seen	س	سـ	ـسـ	ـس
13	Sheen	ش	شـ	ـشـ	ـش
14	Sad	ص	صـ	ـصـ	ـص
15	Dad	ض	ضـ	ـضـ	ـض
16	Tah	ط	طـ	ـطـ	ـط
17	Zaa	ظ	ظـ	ـظـ	ـظ

18	Ain	ع	عـ	ـعـ	ـع
19	Gain	غ	غـ	ـغـ	ـغ
20	Faa	ف	فـ	ـفـ	ـف
21	Qaf	ق	قـ	ـقـ	ـق
22	Kaf	ك	كـ	ـكـ	ـك
23	Lam	ل	لـ	ـلـ	ـل
24	Meem	م	مـ	ـمـ	ـم
25	Noon	ن	نـ	ـنـ	ـن
26	Haa	ه	هـ	ـهـ	ـه
27	Waw	و	و	ـو	ـو
28	Yaa	ي	يـ	ـيـ	ـي

- Some Arabic character have the same shape and differ in the position of dots by which it will be identified, for example characters ن ،ب the two characters have the same shape and identify with one dot, but they differ in position of dot one is, above the baseline (character Noon), and other under the baseline (character Ba), this differentiation can change the meaning of a word.
- Arabic characters of a word are connected a long baseline, and character position above below the baseline. As seen in Figure (2.2) which illustrates the word "samah"; the character ˝Seen˝ appear above the baseline, while character ˝Meem˝ appears below the baseline.

Figure (2.2): Arabic Characters of a Word are connected a Long Baseline

- The width and high of Arabic characters are differ from one character to anther see Figure (02.3).

Figure (2.3): Different Arabic Characters with different Width and High

- The shape of Arabic character varies per the writer, as shown in Figure (2.4) which illustrates the character Khaa written by four writers.

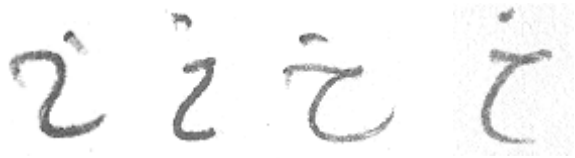


Figure (2.4): same Arabic Character written by Four Writers

- Arabic writing is cursive, most of Arabic characters are connected from two sides; right and left, and only six characters are connected from right side only, as shown in Figure (2.5).

Figure (2.5): Arabic Characters which can be connected from Right to Left

- Moreover, Arabic language has some diacritics called Tashkeel. The names of these Tashkeel: Fattah, Dhamma, Kasra, Sukun, Shadda, Fathatain, Kasratain, Dhammatain also combination of them are

possible. These diacritics may change the meaning of specific word, for example: when we put Fattah diacritic on the word "حر" it became "حَر" which meaning "hot weather", when we put Dhamma diacritics on the same word, it became "حُر" which meaning "free".

- In Arabic language, there are two supplementary characters that operate on vowels characters to create new addition characters. These supplementary characters are (Madda ~) and (Hamza ء). The first one operates only on Alef character. While the second operates on Alef, Waw and Yaa characters as shown in Table (2.3). Moreover Lam ((ل and Alef (ا) characters are combined to create new character (لا)[16].

Table (2.3): Supplementary Characters (Hamza ء and Madda ~) and their position in respect to (Alef', Waw and Yaa) Isolated Characters.

Serial No	Character	Isolated
1	Alef Elmadda	آ
2	Hamza On Alef	أ
3	Hamza Under Alef	إ
4	Hamza On Waw	ؤ
5	Hamza On Yaa	ئ
6	Lam El Alef	لا
7	Madda On Lam El Alef	لآ
8	Hamza On Lam El Alef	لأ
9	Hamza Under Lam El Alef	لإ

- Some Arabic words consists of more than one sub-words. A sub-word is the basic standalone pictorial block of the Arabic writing [17].

- The Arabic words divided to sub-words by one of six characters in Figure (2.5). Also, Figure (2.6) shows an example of words consisting of one sub-word, two sub-words, 3 sub-word and sub-words.

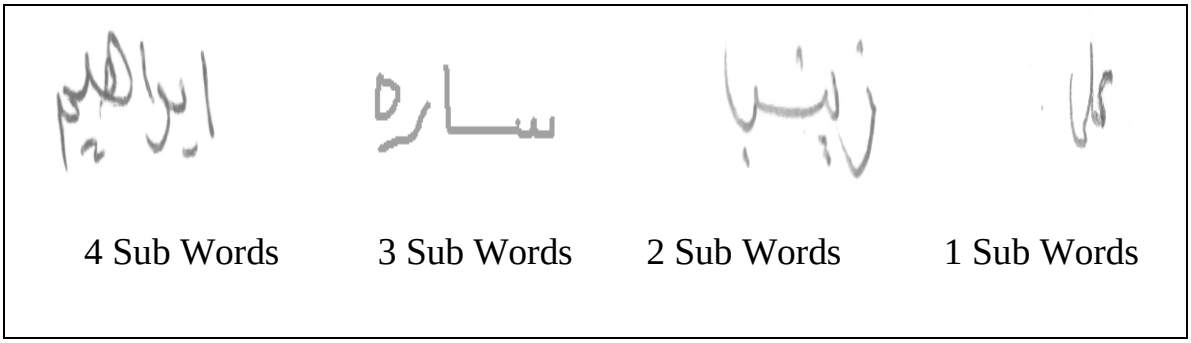


Figure (2.6): An Example of Arabic Words with One or More Sub-Words.

- In addition to all, some Arabic characters can be overlapped with other characters, which made ambiguous meaning. for instance, in the word "احمد" in Figure (2.7), the Haa character" ح" and Meem character are overlapped with each other. Moreover, the character Haa became similar to character "هـ", this due to the variation in writers' style, which causes error in recognition phase.

Figure (2.7): An Example of Characters Overlapping.

A brief details of Arabic handwritten characteristic were reviewed by Lorigo [18] and Ahmed Zaki al[19]. Arabic handwritten characters differ from English, Hindi and Chinese characters in many aspects[20]. As seen,

Arabic characters written from right to left, while other language written from left to right. Furthermore, there are great differences in number of characters; Hindi language contains 40 characters, while Arabic language contains 28 characters, (which is similar to English 26 characters).

Also, there are some vowel characters in Arabic and English languages; 2 vowels and 5 vowels respectively, but not found in Hindi.

Arabic words may be decomposed into sub-words and have 3 complementary characters, which not observed in Hindi language.

Arabic characters may have up to 4 shapes, but are limited in English characters to only two shapes (capital and small) and only one shape for Hindi characters.

Both Arabic and Hindi languages are characterized by distinctive special diacritics, while English language lack such diacritics. Comparisons between these languages are summarized in Table (2.4). More details can found in[8],[21].

Table (2.4): Comparison between Various Languages

Characteristics	Arabic	English	Hindi
Writing Direction	Right to Left	Left to Right	Left to Right
Number of Characters	28	26	40
Number of Vowels	2	5	——
Characters Shape	1 to 4	2	1
Have Diacritics	Yes	No	Yes
Have Sub Words	Yes	Yes	No
Number of Complementary Characters	3	——	——

2.4 The General Recognition System Phases

As mentioned in chapter one, OCR system either can be online or offline. There is no variation between phases of both systems[22]. It depends on lexicon nature, and the recognition approach. The lexicon is a key point to the success of any OCR system[23,24].

As the size of lexicon grows, the recognition efforts and the complexity are increased. So, the general phases of OCR can be described by six phases[25].

First, the data can be captured by several ways depending on the system (online or offline). Then the scanned text image may need to be passed through several preprocessing steps.

After the preprocessing process, the text image may need to be segmented into lines, words, pieces of words, characters or pieces of character. To facilitate the recognition phase, useful features are extracted from the text image, then the valuable classifier methods were used to build the model. Finally, to improve the recognition rate, some post-processing operations may be applied on the model. But post processing phase can narrowly apply and limited to few systems as in [26, 27]. So, the general phases of OCR systems are: data capture, preprocessing, segmentation, feature extraction, classification and, post processing as shown in Figure (2.8). The next sections give some details of these phases.

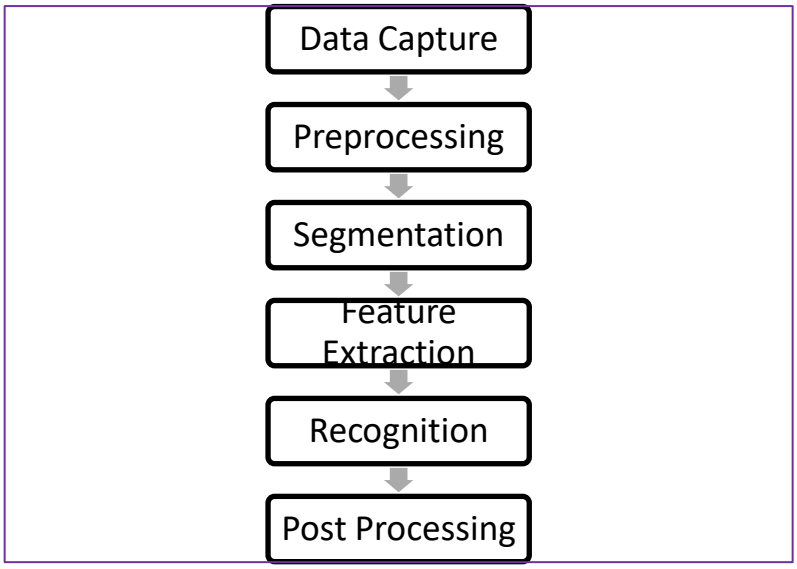


Figure (2.8): The General Recognition System Phases

2.4.1 Data Capture Phase

Data capture is usually depending on the recognition system. In online, it carried out during the writing process by special pen on an electronic interface. While in offline system it being captured usually by optically scanning a paper document. Then the resulting data is stored in a picture file format. In general, the grey-level scanning will be performed at a resolution of 300-1000 dots per inch[28].

In this thesis, samples of Arabic characters' handwritten data, taken from SUST/ALT dataset were used. It is a public dataset used for recognition purpose. A detail description of these dataset can found in (Section 2.6).

2.4.2 Preprocessing Phase

After scanning the text image preprocessing processes must be performed. It is used to remove all elements in the text image that are not useful for recognition process. The main advantage of these phases is to introduce the information in a suitable form, which facilitates the recognition phase.

The accuracy of a recognition system depends on the successful of preprocessing phase. Usually preprocessing phase consists of some operations such as: binarization, normalization, baseline estimation noise detection and thinning [29,30].

After optical scanning process, grey scale images are obtained. Therefore, we need to binarized these images. The aim of the binarization is mainly converting the gray scale image into a binary image based on threshold[31]. These process increases the processing speed and decrease the storage space.

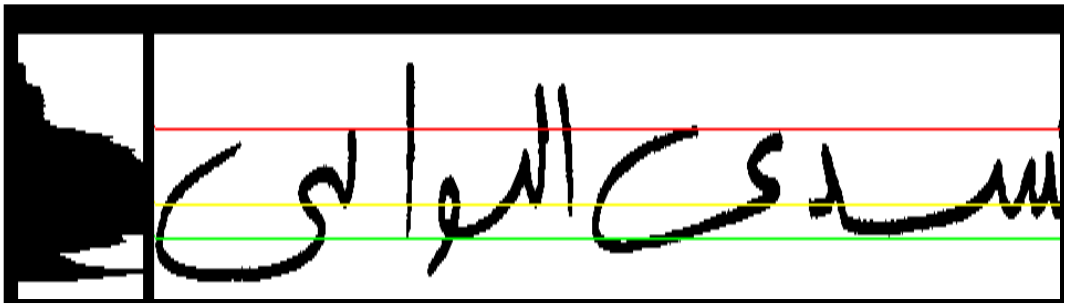
The method used to binarize is known as thresholding. Usually the images are converted in to two levels. These levels are white for background pixels and black for foreground pixels.

Two types of thresholding categories are existing. These types are global and local thresholding. In global thresholding, threshold selection leads to a single threshold value for the entire image[32,33].This value is often based on an estimation of the background intensity level of the using an intensity histogram. In local thresholding different values are used for each pixel according to the local area information[31].

Local thresholding is commonly used in systems that involve images with unstable level of intensities, or for poor quality images. Examples of these images are document images and pictures from satellites camera.

Normalization is the most important preprocessing operation for character recognition. The goal for these operation is to reduce the variation of the shapes of the text image and to reduce the variation of handwriting style. In order, to facilitate feature extraction process and improve their classification accuracy. Several factors in the text image can be normalized, e.g. skew, slope , slant , height and width of the image are normalized in[34,35].

In Arabic language, baseline is defined as line on the character of Arabic word is connected along it and distributed above or below it[36]. It provides valuable information about the text orientation and provides the association points between characters. in some cases, baseline must be estimated. Tt's important to find baseline exactly and to extract correct features along the baseline's position, and over, under baseline[29]. The estimation of baseline is more difficult, so the methods for baseline estimation must take a great attention. The projection methods are widely used for this operation. This is because it is robust and easy to implement F.MENASRI al.[37] used projection histogram to estimate the baseline , using the following steps: First the dot and diacritic are removed. Second to prelocate a histogram band the loops are detected. And finally they compute the projection histogram and estimate the upper and lower baselines as shown in Figure(2.9) .The literature of the projection methods can found at [34,38-39].



Figure(2.9): Upper and Lower Baselines Detection [37]

Noise detection is an important preprocessing operation specially in the field of offline handwritten recognition. This is because scanning devices and transition media introduce noise and unwanted elements.

Also spusrion pixels may present , these pixels are noise pixels that add irregularities to outer contour of the word .Smoothing and thinning operation always are used to remove noise[38].

In addition, the variation of written style may cause noise. In this case

thinning is used also because it facilities the extraction of some features such as loop, cusp points.

Therefore, thinning is an important preprocessing operation for the analysis and recognition of different types of images. One advantage of thinning, it minimized the size of data required to be processed. It also made the analysis and recognition of an image more easily. In addition, it reduced the time of classifier algorithm.

In general thinning algorithms can be classified into two categories[40].

The first category is iterative thinning algorithms. The idea of this algorithm is making several pixel by pixel operations on image, to obtained the suitable skeleton. The most common types of these algorithm are sequential and parallel thinning algorithms. The second category is non-iterative thinning algorithms. Figure (2.10) describes the categories of thinning algorithms

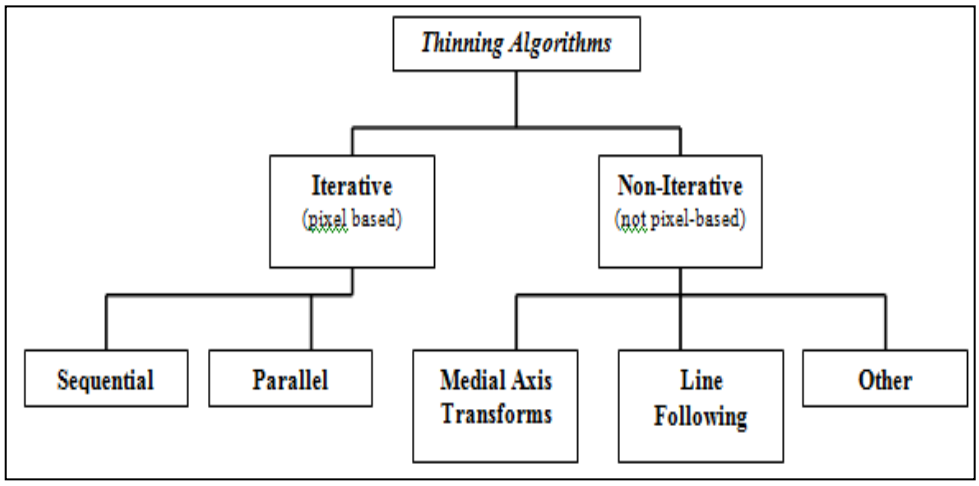


Figure (2.10): Categories of Thinning Algorithms [40]

2.4.3 Segmentation Phase

The segmentation phase is a necessary phase in Arabic handwritten recognition systems. A poor segmentation process leads to miss recognition or rejection object[41]. Although this phase is a necessary phase but it may not require in some systems.

The goal of the segmentation algorithm is to partition a text image into regions or sub-regions[42]. Each region containing an isolated object. There are two techniques have been applied to segment the handwritten Arabic words in to individual characters or sub-word.

Also, these techniques can be used to segment the character into sub character.

The two techniques are implicit and explicit segmentations[43,44], which are discussed in the following subsections.

2.4.3.1 Implicit Segmentation

In implicit segmentation, the words are segmented directly into character using a segmentation algorithm[45].

A fast segmentation method for offline Arabic handwriting based on implicitly segmentation is presented in[46]. First the document is segment into line, then into words, finally into characters.

The segmentation process is implement by projection method and contour following method (to avoid character overlapping). The segmentation process starts at the last character of the words. Because these characters have a large curve which facilitates the segmentation process.

2.4.3.2 Explicit Segmentation

In explicit segmentation words are externally segmented into sub-words which are then recognized individually[47]. These approach is widely used for Arabic handwritten because it is focus on using more features.

Benouareth, A., et al.[38] build a feature vector sequence to describe each word. They used explicit word segmentation. The basic idea of the algorithm is divided the image from right to left into many vertical windows or frames.

They have adopted two segmentation schemes into frames. The first one is uniform where all frames have the same width. The second segmentation scheme is non-uniform see Figure (2.11). In this scheme, the frames do not necessarily have the same width. The boundaries of each frame are based on minimum and maximum analysis of the vertical projection histogram.

This analysis consists in defining the frame boundaries to be the midpoints between adjacent minimum/maximum pairs. These midpoints must verify some heuristic rules related to the distance between the corresponding adjacent minimum/maximum pairs.

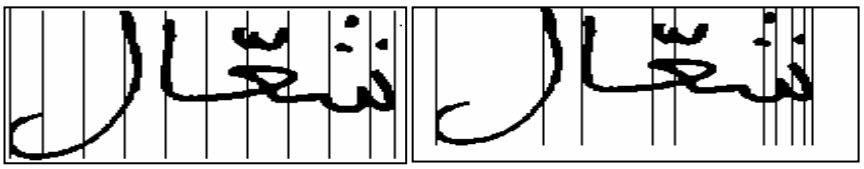


Figure (2.11): Uniform Segmentation and no Uniform Segmentation [48]

2.4.4 Feature Extraction Phase

Many features have been discovered and used in the field of handwritten recognition[49]. They are used to measure the attributes of recognition patterns.

The great difference in the characteristics of Arabic language lead to variation of features. So, any researcher chooses the suitable feature to his system. In other words, features are different from on system to another in the field of Arabic recognition systems.

In general features can be classified into two groups[8]:

Statistical global features which are usually statistical or topological. it used statistical distribution to represent the text image. They used to reduce the complexity, therefore the higher speed can be a chivied.

The common statistical features which used in the field of text recognition are: image zoning, centroid, distances, Fourier transform, invariant moments, vertical and horizontal count etc.

Text image can be successfully represented by extracting many topological features such as the distance between two points, the ratio between width and height of the character, connectivity, number of connected components, number of holes, etc.

Structure features which are usually geometric. These features provide some knowledge about the structure of an image.

Various characters properties can be represented by geometrical features such as concave parts, convex parts, intersections, endpoints, width and number of a stroke, etc.

In this thesis, we used Freeman chain code methods to detect the endpoints of characters. Figure (2.12) shows some of structure features.

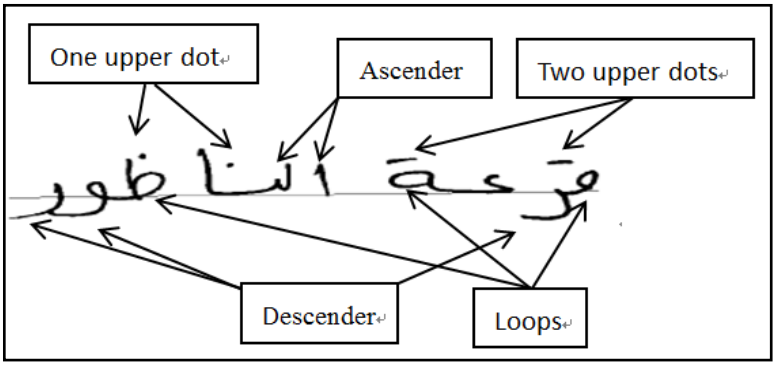


Figure (2.12): Structural Features of Tunisian Town Name "قرعة الناطور" that contain Two Words and Seven Paws [50]

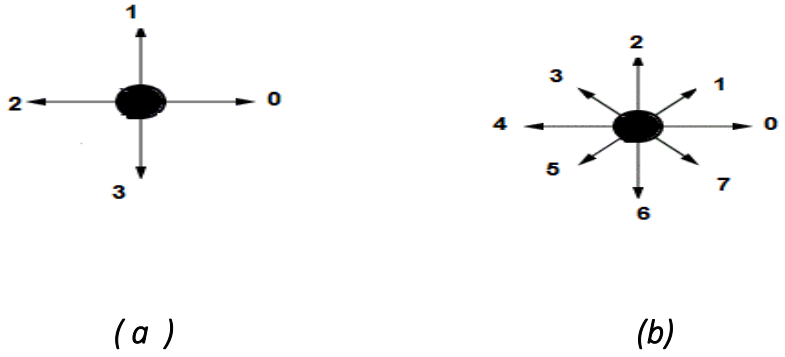
Several feature extraction methods are now available [51-53].An excellent survey on features extraction methods can be found in[54]. One of the most popular method is Freeman Chain Code(FCC). It used to extract robust topological features.

Freeman Chain codes (FCC) was introduced by Freeman in 1961[55].Then they became one of popular method for shapes representations.

This method used to specify the boundary of a shape by a connected sequence of straight line segments and trace the length and direction of an object.

It can be start form any boundary pixel and then moved to next neighbor pixel and so on until the starting pixel is reached. The move may be clockwise or counterclockwise. There are two types of(FCC) :4-connectivity and 8-connectivity.

The 4-connectivity FCC has used four graphic direction to find the border of the shape and assign an integer number ranged from 0 to 3 corresponding to the four original directions (E, N , W , S) .Otherwise 8-connectivity FCC represented by 0 to 7 number coded corresponding to (E, NE , N , NW , W, SW , S, SE) directions. The two types are illustrated in Figure (2.13 a) & Figure (2.13 b)



(a) (b)

Figure (2.13):4-Connectivity and 8-ConnectivityFreeman Chain Code

Freeman in [56] used a chain code of the boundary to extract the critical points. Then he used these critical points to produce a shape description that is invariant to translation, rotation, and scale.

2.4.5 Classification Phase

In the classification phase, again several extra features can extract from text image in order to classifier it.

The selection of the method for classification is an important step for achieving better recognition rate. Over the time many methods have been explored by researchers to classifier (recognized) the Arabic handwriting. The most common methods are[8] :

- Statistical methods, according to the class-conditional probability density estimation approach, statistical methods are divided into parametric and nonparametric ones [57].
- Structured and syntactical methods.
- Machine learning methods.
- Expert systems methods.
- Neural networks methods.

Every method has many advantages and disadvantages, which leads for using hybrid approaches for different methods. Some of this works are summarized in this section.

Neural networks(NN) is widely used for problem that not explicitly formulated and very useful method to learn the handwritten characters. Several recognition systems have been models base on this method.

For instance, Leila, C. and Mohammed,B [58] presented Arabic offline handwritten recognition system based on Fuzzy ART(type of neural network) their system based on an holistic method includes four phases.

Preprocessing phase consists of thinning, normalization, slant detection and slant correction operations.

The tchebichef geometric moments feature is extracted base on discrete orthogonal Tchebichef polynomial. Two Fuzzy ART s algorithms are constructed for training and classification phases. The proposed system has been applied on IFN/ENIT dataset, and achieved a nearly 78% recognition rate.

Graves, A. and Schmidhuber,J. also used multidimensional recurrent neural networks to recognized offline Arabic word as in[59]. Their system achieved about 91.4% accuracy rate.

Due to successful of implementation of Hidden Markov Models (HMMs) in speech recognition, many researchers have used them for offline Arabic handwritten recognition.

Especially in limited dataset, because it gives robust results[60].
Märgner,V.,et al.[61]presented an offline handwritten Arabic word recognition system by using HMM as classifier method and based on character recognition without explicit segmentation.
The system applied on IFN/ENIT dataset. Therefore, all fundamental preprocessing tasks are already done.
The researcher starts the preprocessing from binary images of towns name. First a contour of the image is represented by contour coding algorithm. Then noise is reduced from the contour list by deleting contours that are too short to be part of a character. Then skeletonisation is applied on the list of the contour representation. Then word is normalized into constant average character width, image is normalized to grey level word image as final step in normalization.

To extracted the features, sliding window method is used with three columns and works from right to left. The result is one feature vector which is reduced by Karhunen-Loève Transform (KLT) method to obtain the relevant features only.

HMM classifier has been used with 7 states and 3 transitions (recursive self-transition, transition to the next state, and a transition that skips the next state). Word is segmented into characters automatically by a Dynamic Programming clustering procedure (n x 7 segments)., then the initial segments are obtained by using the back-propagation method.

To initialized the model, first codebook has been initialized by LBG-algorithm and optimized by EM-algorithm. Viterbi Algorithm is used to training of the HMM parameters. Viterbi Algorithm is used also for recognition phase, a tree structured lexicon and beam search strategy are also implemented to avoid the time-consuming problems.

The system has been applied to IFN/ENIT dataset and achieved better results.

Another study presented by Alma'adeed, S., et al.[34], they used HMM classifier to constructed an Arabic handwritten word recognition system. First, the word image is cropped, then the slant and slope of the word is corrected and thinned. Segmentation process is implemented to segment the words into characters or sub-characters.

The useful features are obtained from the segment images to build feature vectors, which were then partitioned into several groups of, this information was used to train an HMM by using Forward-Backward algorithm for every word, using the.

Also, preprocessing, feature extraction, and segmentation processes were done to give observation vectors. So, all HMMs parameters are defined, then the log-likelihoods for the HMMs is calculated, the word assume to be recognized if log-likelihoods is highest.

The system has been applied to a dataset consists of 10000 Arabic words, and achieved a nearly 45% recognition rate.

The higher recognition results can be achieved by combining different recognition methods or classifiers.

For instance AlKhateeb, J.H., et al.[51]combined K-nearest neighbor (kNN) and neural network (NN) methods to recognized of unconstrained handwritten Arabic words.

Preprocessing was performed by using skeletonization technique, to remove the variation in the handwritten word. Edges of the words images were extracted by horizontal high pass filter, DCT (Discrete Cosine Transform) features, Wavelet features, the moment invariant features, are also extracted.

The first classifier was KNN, it predicated the recognition word by computing the minimum the Euclidean distance between training and testing data. multilayer perception back propagation neural network with the input layer, hidden layer, and output layer, is used as second classifier. The system was training And testing on IFN/ENIT dataset, sets (a,b,c,d) for training and set e for testing. The recognition rate was good.

Chergui Leila et al.[62] presented multiple classifier recognition system for Arabic handwriting (Multi-Layer Perceptron “MLP”, Radial Basis Function “RBF”, Fuzzy ART) to improve the accuracy rate from 84,31% to 90,10%.

HMMs can be also combined as in[63]. They combined three homogeneous HMM classifiers with the same topology (right to left) and different only in the orientation of the sliding window. They aimed to recognize offline handwritten Arabic city names. The authors tested their method on benchmark IFN/ENIT database of Arabic Tunisian city names. They achieved better recognition rate about 90%.

2.4.6 Post-processing Phase

This is the final phase in the OCR system. Classification phase sometimes produces several solutions, therefore post-processing phase is needed. The task of this phase is selecting the unique right solution from the possible set of solutions[28]. Also the solution of classification may contain some recognition errors, so post processing methods remove these errors by dictionary lookup(top down) and statistical approaches (bottom up)[64].

Taghva, K.,et al. [65] designed post-processing system for automatically correcting error caused by OCR system devices.

2.5 The Main Approaches for Text Recognition

There are two main approaches that can be used to develop offline and online Arabic handwritten for recognition system. The first approach called holistic approach while the second approach called analytical approach[66]. In this section, the two types will be discussed.

In holistic approach, the system recognizes the whole image (character or word) as one object without segmentation , and this being done by using holistic features such as rotation, word length , ascender ,bounder etc.[67].

Wafa Ali and Musa,M.[68] implemented an Arabic handwritten names recognition by using holistic approach. They designed a probabilistic neural network to recognize the popular Arabic names holistically. They used SUST (Sudan University of Science and Technology) dataset to test the system performance. Their system achieved well recognition rate.

In analytical approach, the system does not recognize the image as whole, instead the image segmented into pieces as character or sub-word. Then segments are combined to match the models. Therefore, the analytical approach basically has two phases. These phases are segmentation and combination[8],[69].

Analytical approaches can be divided into three groups[70]:

- Character-based approaches that recognize each character in the word and combine the character recognition results using either explicit or implicit segmentation as shown in (section 2.4.2).
- Grapheme-based approach which uses graphemes such as structural parts in characters (e.g., the loop part in character, arcs, etc.) instead of characters to minimize the matched units.

  Sari,T., et al. proposed new segmentation algorithms for offline handwritten character based on morphological rules (about nine rules) [70]. The operation accepts or reject the Local Minima (LM) in the lower outer contour of the sub-word as valid segmentation points (VSP).

  F.MENASRI al. Deigned shape-based alphabet for offline Arabic handwriting recognition system[37]. The system based on explicit grapheme segmentation.

  - Pixel-based approach that uses features extracted from pixel columns in sliding window to form words models for word recognition. For example Pechwitz,M. and Maergner,V. designed a segmentation algorithm for training their system[71].

  These algorithms base on extracting features from pixel values. Then the features organized into vectors or frames. Features are extracted by using sliding window with three columns.

  Comparing the two methods, we can be observed that the holistic models must be trained for every word in the lexicon, while analytical models need only be trained for every character in the lexicon, and this is limit to few models.

  Therefore, holistic approach limited to small and constant lexicons, such as recognizing person's names, city names, companies' names and bank chequs …etc. On the other hand, holistic approaches are easier to

be implemented with better recognition rates. This is because all words in the lexicon are trained.

2.6 Offline Arabic Handwriting Datasets

The most important issue for the development and comparison of recognition systems is how to find a large database with truth information. Therefore, standard datasets for offline Arabic handwritten are needed, to facilitate the recognition phase.

Several datasets are used for this purpose. Many researchers need to found grouped information about the freely dataset. Therefore, in this section we tried to discuss some commonly datasets.

Since isolated characters from SUST/ALT dataset will used to training and testing our system, we described these datasets in detail in (Section 2.6.1). In (Section 2.6.2) information and details about IFN/ENIT are presented, because it used by many researchers and contains words and characters. Also, IFHCDB datasets described in (Section 2.6.3).

2.6.1 SUST/ALT Dataset

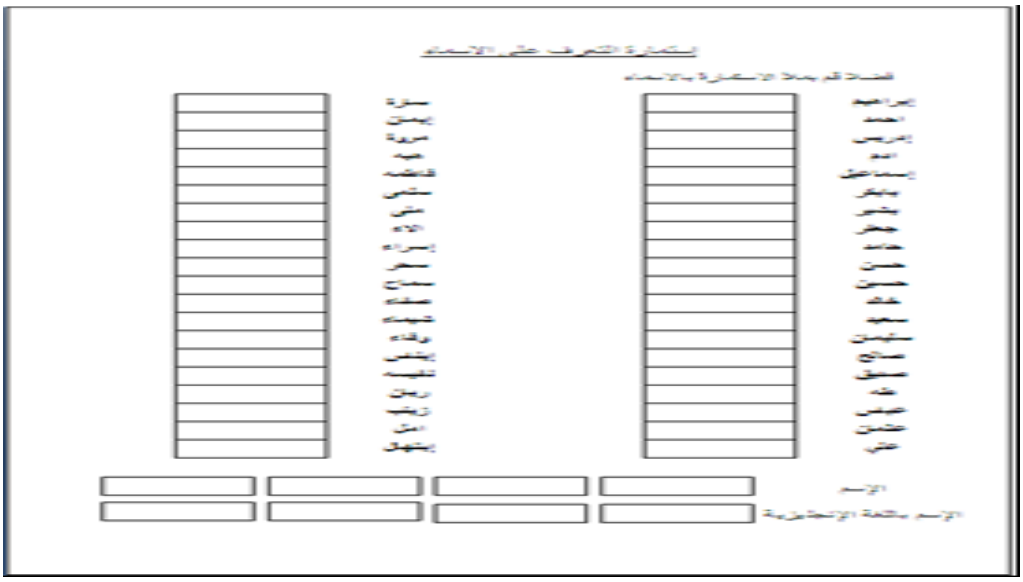
SUST/ ALT Dataset SUST dataset, it’s a new dataset developed and published by SUST/ALT (Sudan University of Science and Technology-Arabic Language Technology group) group. It contains numerals datasets, isolated Arabic character datasets and Arabic names datasets[72].

Since there are few Arabic databases available, it become a popular dataset and used in various studies[73-75].

It contain Forty common Arabic (especially in Sudan) males and females' name[76].

Each form written by one writer resulting 40,000 sample. After data collection process, each name is keeps into separate folder. These represent

the word form, it used for researching purpose.



Figure(2.14): SUST Names Dataset Filled Form [76]

As appear in Figure (2.14) the form contain pre- typing Arabic names and text boxes left to the writer to repeat the names by his hand. In the bottom of the form the writer was asked to write his full name in Arabic and English language.

SUST /ALT group also, design new dataset from the names dataset. This new dataset represents the isolated characters. We used it to test and train our suggested system.

The objective of this dataset is to introduce an offline isolated character images, to use by different researchers for classification purpose. SUST group, designed a form as illustrated in Figure (2.15) to collect the required isolated Arabic characters.

One hundred and forty-one forms have filled by different writer. Then, these forms scanned by scanner device accuracy of 300 dpi and saved as bmp images[77]. This process extracts each specific character from all form and put it in a separate folder as grayscale image, any character composed about 1411 images written by hand as shown in Table (2.7).

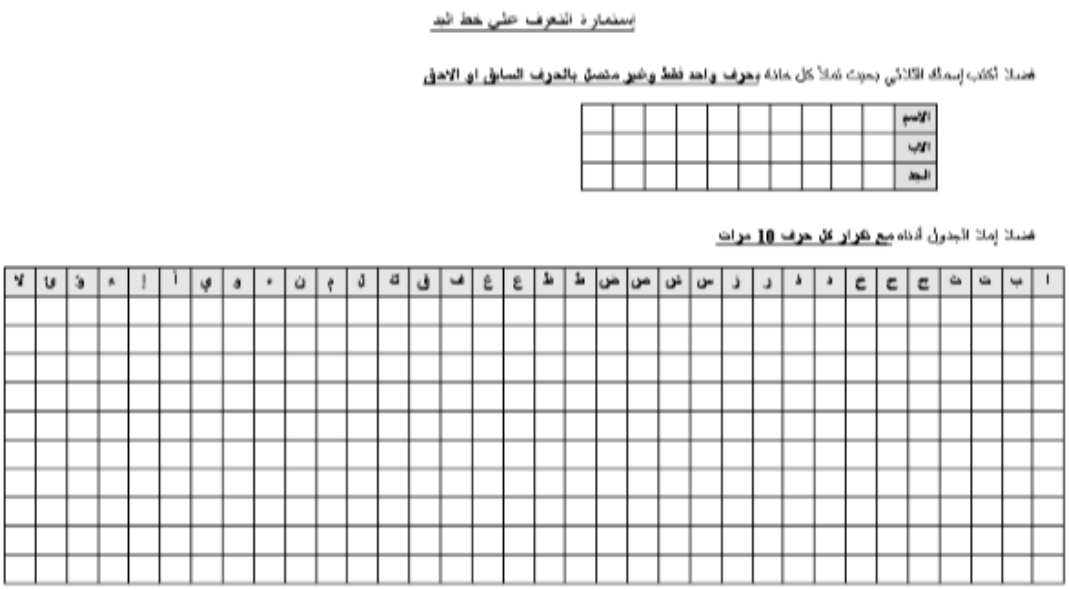


Figure (2.15): SUST Form for Isolated Arabic Characters

To increase the data quality, it was stored in BMP format. Each of black-and-white bitmap image displays as one of the 34 characters in Arabic language (28 basic characters plus 6 ligatures characters). Each character within these 34 characters was scaled to 8.61 KB. The high and width was fixed to 88 pixels for each character. The dataset information is summarized in Table (2.6).

Table (2.5): SUST Dataset Information

Dataset Creator:	**SUST/ALT group**	**Number of Instances Categories**	**34 categories (characters)**
Date Donated:		**Instance Type:**	Bitmap Image
Area:	Computer	**Instance Size:**	8.61 KB
Associated Tasks:	Classification	**Instance High:**	88 Pixels
Number of Instances:	47,988	**Instance Width:**	88 Pixels

SUST isolated Arabic characters' dataset contain about 48,000 samples

for 34 isolated Arabic characters, each isolated character kept in separate folder hold the name of that character. Every character image is saved with a1` name and number indicating its writer. As mentioned the dataset are bitmap image, to increase the data quality it was stored in BMP format. Table (2.7) below shows the characters and the frequency of each character.

Table (2.6): SUST Isolated Characters Dataset

Serial No	**Name**	**Character**	**Frequency**
1	Alef	ا	1208
2	Baa	ب	1410
3	Taa	ت	1410
4	Tha	ث	1410
5	Jeem	ج	1410
6	Hah	ح	1410
7	Kha	خ	1410
8	Dal	د	1411
9	Thal	ذ	1407
10	Raa	ر	1410
11	Zay	ز	1411
12	Seen	س	1410
13	Sheen	ش	1410
14	Sad	ص	1410
15	Dad	ض	1410
16	Tah	ط	1410
17	Zah	ظ	1410
18	Ain	ع	1410
19	Gain	غ	1410
20	Faa	ف	1410

21	Qaf	ق	1410
22	Kaf	ك	1411
23	Lam	ل	1410
24	Meem	م	1410
25	Noon	ن	1410
26	Haa	ه	1410
27	Waw	و	1410
28	Ya	ي	1410
29	Lam Al Alef	لا	1410
30	Hamza	ء	1410
31	Hamza On Alef	أ	1612
32	Hamza Under Alef	إ	1410
33	Hamza On Naberah	ئ	1412
34	Hamza On Waw	ؤ	1408
Total			47,988

2.6.2 IFN/ENIT Dataset

Many researchers implemented their system on IFN/ENIT, which introduced by Pechwitz, M., et al. in[78]. It developed by the Institute of Communications Technology (IFN) at the University of Braunschweig in Germany and the Ecole Nationale d'Ing'enieurs de Tunis (ENIT) in Tunisia[79].

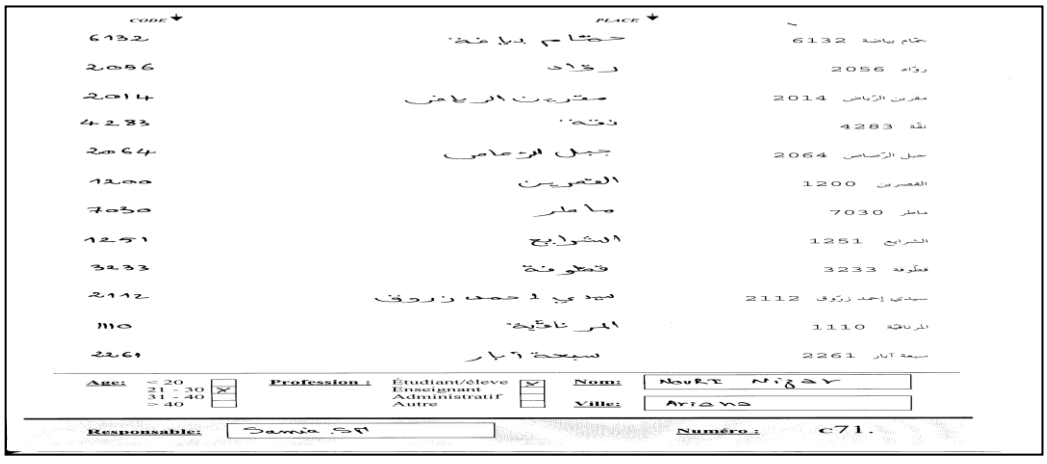
It is freely and open dataset. Freely means it not for commercial purpose. open means it available with no guarantee access for all users, any person need only to visit http://www.ifnenit.com/ web site and download it. It used by several researchers for recognition purpose[80-83]. It contains Tunisian towns/villages name.

First, they constructed a form. then asked 411 writers to fill it by writing the required information. The form divided in to four areas: The right area consists of 12 lines with printed Tunisian town/village names and their postcodes. The middle and left areas are designed for writers to hand write the town/village names and their postcodes. Finally, the bottom area designed for the writer to write his personal information such as name, age and profession (an example of the filled form is shown in Figure (2.16). Each writer asked to fill five pages in order to write 60 town/village names. The dataset consist of about 26400 word containing more than 210000 characters[78].

The complete dataset is divided into four sets (a, b, c, d) to use for training and testing offline Arabic handwritten systems. The number of town/villages name are 937. These names may be single word such as " قطوفة", or composite word "أولاد حفوز".

All data appear in tiff and bmp format. All images are digitized using a resolution of 300 dpi.

The dataset also contains ground truth information(GT) e.g. series of Arabic character shapes, top line sets position, baseline position and reference position. The ground truth information was stored on ASCII .txt file.



Figure(2.16) : An Example of IFN/ENIT filled Form [78]

2.6.3 IFHCDB Dataset

IFHCDB (Isolated Farsi Handwritten Character Data Base), is character dataset that designed by Mozaffari,S.,et al.[79] for isolated Farsi handwritten characters , but it can be used to recognized Arabic character. Because Farsi characters consist of 32 characters ,28 of them are similar to Arabic characters plus 4 characters special for Farsi language.

As shown in Figure (2.17) the 4 special Farsi characters are marked by *. The dataset consists of 52,380 isolated characters and 17,740 numerals. The numerals are same as Arabic numerals 10 numerals (0-9). All the data were gathered from real handwritten documents (Iranian high school and guidance school entrance exam forms during the years (2004-2006).

This dataset is useful for researchers whose works in pattern recognition, because it divided into two parts one set for training 70% and the second set for testing 30%. This dividing made to put the data into comparison fashion. All the images were stored as 77×95 BMP image. These images are grayscale. Also for each grayscale image the meta data (10 attribute) are stored into text file.

This meta data included some information about the dataset and writers, such as file name, writer city and writer full name. The dataset is completely free and available for academic researcher at

http://ele.aut.ac.ir/ imageproc/downloads/IFHCDB. html web page.

16	15	14*	13	12	11	10	9	8	7*	6	5	4	3*	2	1
ش	س	ژ	ز	ر	ذ	د	خ	ح	چ	ج	ث	ت	پ	ب	ا
32	31	30	29	28	27	26*	25	24	23	22	21	20	19	18	17
ی	ه	و	ن	م	ل	گ	ک	ق	ف	غ	ع	ظ	ط	ض	ص

Figure(2.17) : An Isolated Farsi Character Set [79]

2.7 Isolated Arabic Handwritten Characters Recognition State of the Art

This section discusses the developing of offline system for an Isolated Arabic handwritten characters. The following trials do not include developing online systems.

Recognition of an isolated Arabic handwriting characters, is an important area of research, there are a big challenge of this area. many researchers have worked in the field of isolated character recognition using various methods[4, 32, 84, 85]. These works are summarized below.

El-Glaly,Y. and Quek,F designed[86] two classifiers to recognize an isolated handwritten Arabic characters. These classifiers are back propagation neural network(BPNN) and the k nearest neighbor algorithm(KNN).

First thinning process is done to remove noise from whole image. They used Otsu's method to convert the grey images into black and white images.

Then all the images are resized to be in same size.14 features are extracted from each character image (4 from the whole image and 10 from image segment).

For neural network classifier, they built network with two layers with 14 inputs and 28 neurons in its output layer to identify the letters. The second classifier is also built to classify a new character, the system finds the k nearest neighbors from the training datasets, and uses the categories of the k nearest neighbors to weight the category candidates.

The two systems implemented by MATLAB software, then tested with 196 characters for training and 84 characters for testing. Results show that kNN classifier work better for training datasets with 100% accuracy rate, has very low error rate in testing datasets, also it not sensitive to noise in

the input images. On the other hand, BPNN has high error rate in the training data, and very low accuracy in the testing data, and it is sensitive to high noise.

Abed, M.A., et al. [17] proposed an OCR system for handwritten isolated Arabic characters. Freeman chain code has been used as classifier. The system was trained and validated on 7400 Arabic characters with 18 forms.

The researcher ignore the reminder 10 character from the Arabic language because for similarity of Arabic character for example the body of (ب) Baa character is similar to the body of (ت,ث) Taa ,Thaa characters so only the body of this characters is taken this constructed one character , and so on other characters the result is 18 characters instead of 28 characters.

Then the coordinates of the boundary pixels are obtained and Freeman chain code obtained from these coordinates and normalized. To recognized the characters the normalized Freeman chain code (for 18 characters) is compared with the existing database, if match occurs then the character is recognized.

The system implemented by MATLAB software and achieved 95% accuracy rate.

Rachidi, A.,et al.[87] presented an automatic system to recognized handwritten isolated Arabic characters based on criterion constructed from pre topological pseudo distance.

First, they constructed the set of each pixel in image and it is eight neighbors, interior, frontier, border and coherency. So, the pre-topological are defined by these elements. To measure the stability of a class of characters pseudo distance (the length of the shortest bath between two

points) is used, the distance between two similar character are obtained, if this distance is small the character are recognized otherwise the character is rejected. This method tested and trained on a dataset consists of 6188 handwritten isolated Arabic characters and achieved robust results.

Abed, M.A.,et al. [88] design new approach to simplify characters handwritten recognition based on genetic algorithm. Some preprocessing operations are done. Feature extracted from the input images then compared with saved template's features. In this system, experimental results display the higher degree of performance about 95%.

Abandah,G.A.,et al.[89] proposed multiple classifiers system to recognized character form. The system has been applied to a dataset consists of 30 isolated characters, 22 initial characters, 22 medial characters and 30 final characters.

Features are extracted from the secondary part of the letter (disconnected from the main body), main body row, main body skeleton and main body boundary. Researchers built five classifiers to recognized the character forms. These classifiers are QDA(Quadratic Discriminant Analysis) , LDA(Linear Discriminant Analysis), DQDA(Diagonal Quadratic Discriminant Analysis), DLDA (Diagonal Linear Discriminant Analysis) and kNN (k-nearest neighbor). Results show that initial and final forms are easier to recognize than medial and isolated forms, the best recognition rate was achieved by LDA classifier.

Saad, M.K. and Ashour,W [90]design recognition system for offline Arabic handwritten characters which based on structural, statistical and morphological features from the main body and the secondary components of the character.

First the input image is converted to binary image based on Otsu’s

thresholding method. Then the slant is corrected for every character image, the width and height of character are normalized. To smooth the character's images statistical and morphological noise are removed.

Structural, statistical and topological features are extracted and normalized. Back propagation Neural Network classifier is used for training and testing the data. The dataset(CENPRMI) consist of 308 characters and the system performance was well.

2.8 Conclusion

This chapter has introduced the concepts of OCR and its main approaches and techniques. Also, the main characteristics of Arabic language are discussed briefly in this chapter.

The chapter gave more emphases for OCR and its phases. The OCR phases are divided in to six phases: data capture, preprocessing, segmentation, feature extraction, recognition and, post processing. The preprocessing phase for the scanned image can be divided into five steps: binarization, normalization, baseline detection, noise detection and thinning.

The feature extraction methods are the key success for the classification phase, and they depend on many factors.

The chapter surveys related works on isolated Arabic character recognition, we can conclude that all recognition system phases need more research work. This survey shows that great efforts have been made to advance the accuracy of OCR for Arabic characters. This work spread over preprocessing, feature extraction and classification methods. However, the surveys have shown many gaps that needs more research work.

Another observation that there are no standard methods for features

selection. Therefore, features selection depends on the application nature, the complexity of dataset, the recognition approaches, and the classification methods. Another observation that the segmentation phase is not necessary for some recognition systems. Moreover, the post processing phase increase the accuracy rate and enforce the overall system, but it not applicable in all researches. Final observation that SUST/ALT dataset is a big chance for researches purpose.

CHAPTER THREE

HIDDEN MARKOV MODEL

3.1 Overview

In This chapter, the fundamental concepts of HMM are introduced. The notation of HHM method has been discussed to give an idea about the mathematic equations that will use in this thesis.

Then the three fundamental problems related to HMMs and their efficient solutions are discussed in (Section 3.4). Since the HMM has been used as main classifier method in this thesis, a further review of research done in this area has been discussed and compared at this chapter.

Also, this chapter discussed the application of HMMs on Arabic OCR, speech recognition, phoneme recognition, offline signature, and protein domain recognition.

Most of information given in this chapter is take from Rabiner[91-93] for more details and tutorials readers can refer to these sources.

3.2 The Hidden Markov Model Definition

Hidden Markov Models (HMMs) is a statistical method that uses probability measures to model sequential data represented by sequence of observation vectors. The theory of Hidden Markov Models (HMMs) was first introduced by Baum et al[94].

It provides a powerful statistical framework for solving many applications in artificial intelligence(AI), pattern recognition and speech recognition[95].

It offers several advantages for handwriting recognition, these advantages are:

- Preserving time and correctness of text since segmentation is not

- Required in HMM.
- Availability of several free HMM tools.
- Automated algorithms exist for training the HHM models.
- The theory Behind the Hidden Markov Model method is
- Straightforward and easy to understand.
- Features selection are language independent, in other words the
- Same features can be used for different languages.

Rabiner, L.R defined HMM as a finite state machine having fixed number of states. The states of the model cannot be observed directly (hidden), only the output symbols of the state can be observed. It can be classified as discrete or continuous HMMs according to the output symbol[91].

In discrete HMMs every state has its own discrete probability distribution for each sample, the outputs may be characters from the dataset or vectors from a codebook.

In continuous HMMs the emission probability distribution for symbols is continuous in each state and can be represented by a Gaussian mixture model[96].

3.3 Elements of Hidden Markov Model

According to Rabiner, L.R [91] notations, HMM can be defined by the following elements or characteristic:

- $T =$ length of the observation sequence.
- $N =$ number of states in the model
- $M =$ number of distinct observation symbols per state
- $S = \{S_1, S_2, \dots, S_N\}$ distinct states of the Markov process

- Q_t = the state at time t
- $V = \{v_1, v_2, \dots, v_M\}$ set of possible observations symbols
- $A = \{ a_{ij}\}$ {3.1}

state transition probabilities where

$$a_{ij} = P\left(q_{t+1} = S_J \,|q_t = S_i\right),\ 1 \le i, j \le N \quad \{3.1.1\}$$

- $B = \{b_j(k)\}$ {3.2}

observation symbols probability in state j where

$$b_j(k) = P\left(v_k \ at \ |q_t = S_j\right),\ 1 \le i \le N,\ 1 \le k \le M \quad \{3.2.1\}$$

- $\pi = \{\pi_i\}$, $1 \le i \le N$ {3.3}

From the above equations, the HHM can be used to give the following observation sequence

$$O = (O_1, O_2, \dots, O_i)\ ,\ O_i \in V,\ 1 \le i \le T \quad \{3.4\}$$

As you seen the HHM models dependent on A, B ,π matrices , therefore the HMM represents by λ parameter , where $\lambda = (A, B, \pi)$

3.4 The Three Problems of HMMs

There are three basic problems needed to be solved in HMMs to be implemented in real world applications[97]. These problems are: evaluation, optimal state sequence and training. The three problems and their solutions will be discussed in this section.

3.4.1 The Evaluation Problem

The evaluation is the first problem in the HMMs. It focuses on how to compute the probability that the observed sequence $P(O|\lambda)$ was produced by the given model $\lambda = (A, B, \pi)$, or in other words given the observation

sequence $O = (O_1, O_2, \dots, O_T)$, and a model $\lambda = (A, B, \pi)$ how do we efficiently compute the probability of $P(O|\Lambda)$, which represents the probability of the observation sequence, given the model.

The most suitable solution to this problem is enumerating every possible state sequence of length T to do this, considering the following equations:

$$Q = (q_1, q_2, \dots, q_T) \qquad \{3.5\}$$

where q_1 is an initial state then

$$P(O|Q,\Lambda) = b_{q_1}(O_1).b_{q_2}(O_2) \dots b_{q_T}(O_T) \qquad \{3.6\}$$

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O/Q,\lambda)$$

The probability of a state sequence *Q* can be written as

$$P(Q,\lambda) = \pi_{q_1}.a_{q_1q_2} \dots a_{q_{T-1}q_T} \qquad \{3.7\}$$

we can obtain the probability of $P(O|Q,\Lambda)$ (the probability that *O* and *Q* occur concurrently) by producing the above two equations as following:

$$P(O,Q|\Lambda) = P(O|Q,\Lambda).P(O,\Lambda) \qquad \{3.8\}$$

The probability of O is obtained by the join probability over all possible state sequences of q.

$$P(O,\Lambda) = \sum_{all\ Q} P(O|Q,\Lambda).P(O,\Lambda) \qquad \{3.9\}$$

$$= \sum_{q_1,q_2,\dots,q_T} \pi_{q_1}.b_{q_1}(O_1).a_{q_1q_2}b_{q_2}(O_2) \dots a_{q_{T-1}q_T}b_{q_T}(O_T)$$

What worth noticing, is that the above-reviewed equations are complex. They require a lot of calculations, it involved $2T.N^T$ calculations (there are N^T possible state sequences and 2T calculations required for each state sequence). So, to solve this problem, the forward procedure can be used to reduce the equation to N2.T calculations[8].

The forward variable $\alpha_t(i)$ defined as following:

$$\alpha_t(i) = \pi_i b_i O_1 \qquad \{3.10\}$$

where $1 \le i \le N$

by using mathematical induction, we can compute

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(j)\, a_{ij}\right] . b_j(O_{t+1}) \qquad \{3.11\}$$

where $1 \le t \le T\text{-}1$ and $1 \le j \le N$

then the probability of O is obtained by the following equation

$$P(O, \Lambda) = \left[\sum_{i=1}^{N} \alpha_t(i)\right] \qquad \{3.12\}$$

When we examine the calculations of the forward procedure, we see that it required $N^2.T$ calculations.

In similar way, we can compute the backward procedure (it will be used as solution to training problem), by the following equations.

The backward variable

$$\beta_t(i) = 1 \qquad \{3.13\}$$

where $1 \le i \le N$

Again, by using mathematical induction we can obtain

$\beta_t(i) = \sum_{j=1}^{N} a_{ij} . b_j(O_{t+1})$ where t=T-1, T-2,…,1 and $1 \le i \le N$

the computation of backward procedure, will required $N^2.T$ calculations (same as in forward procedure).

3.4.2 The Optimal State Sequence Problem

The second problem is focusing on finding the optimal state sequence associated of a given observation sequence (how to discover the hidden state of the model), on the other word if the observation sequence

$O = (O_1, O_2, \ldots, O_T)$ and λ model are given, then how do we choose a corresponding state sequence $Q = (q_1, q_2, \ldots, q_T)$ that is optimal in some sense.

Evaluation problem have specific solution while optimum solution is chosen out of multi-solution. Thus, it is difficult due to the definition of the optimal state sequence.

There are several possible optimum solutions to this problem, out of which, the commonest solution is Viterbi algorithm, which finds best sequence for the given observation sequence. The theoretical of Viterbi algorithm can be found in[94] ,[98].

3.4.3 The Training Problem

The third problem of HMMs, is finding the appropriate method for adjusting the parameters (A, B, π) to maximize the probability of the observation sequence given the model P(O/λ).

The Baum-Welch algorithm is an attractive method that used the forward and backward procedure to adjust the model parameters (A, B, π) to maximize $P(O|\lambda)$ [94, 99].

3.5 The Topologies of HMMs

HHMs can be classified according to the structure of the transition matrix to three categories[100]. These categories are:

Fully connected HMMs(ergodic) model.

a) Left–to–right HMMs model.
b) Hybrid HMMs model.

The three models are discussed in the following subsections, for more information reader can refer to ([91],[8]).

3.5.1 Fully Connected Model

In this model, every state should be reached from every other state of the model in a finite number of steps. It must satisfied the following two constrains[8] :

- all a_{ij} coefficient is positive, where a_{ij}'s represents the elements of transition matrix.
- There is no distinguishing between starting and terminating states. That is means every state can be reached from every other state in the model.

This type is useful in speak applications, because they satisfied the conditions. Figure (3.1) illustrates a fully connected HMMs with three states.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

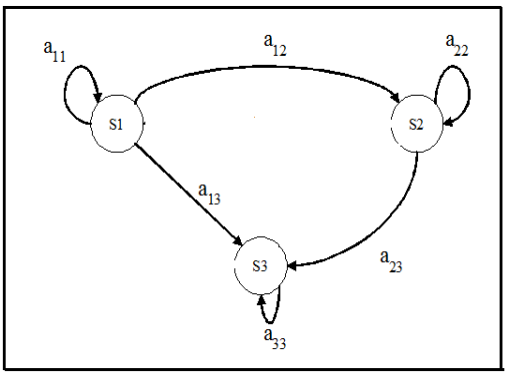


Figure (3.1): A fully Connected HMMs with Three States

3.5.2 Left to Right Model

The fully connected model it is not applicable for all application. In some cases, the left to right model can be used. The key idea of this model is that the state growth from left to right with exception to the loops.

As the time, the index increases corresponding to increasing in the states.

Thus, the left to right model must satisfy the following constrains:

- No transaction allowed to state with indexes lower than the current state index.
- The large jump from one state to other state is not allowed.
- The model begins from start state with lower index and ended with terminating state which has high index.

Figure (3.2) illustrates a left to right model with four states. We will describe the details of this topology in chapter four, because it used in this thesis.

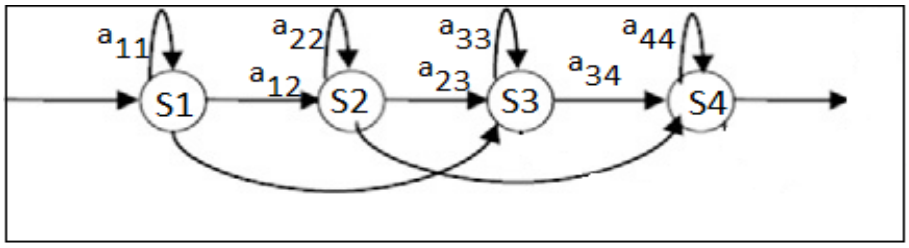


Figure (3.2): A left to Right Model with Four States

3.5.3 Hybrid Model

The combinations between the two topologies can be allowed in some practical cases for example Figure (3.3) shows a cross joined of two parallel left to right model (four states in each model). They constructed new model which satisfied the constrains of fully connected model and left to right model.

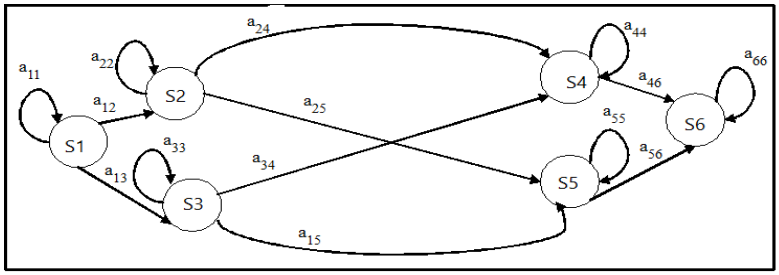


Figure (3.3): A cross Joined of Two Parallel Left to Right HMM

3.6 Applications of HMMs

HMM has been used successfully to model many applications. These applications included speech, text recognition, phoneme recognition, offline and online signature, protein domain identification, etc.

Various works done on these fields, will discuss in the following subsections.

Methods of HMM are applied to speech processing in several researches , Rabiner proposed many studies in speech recognition field and introduced a good theory and tutorial of HMM , he illustrated some application of HMMs in speech recognition in[91],[92],[101] his tutorial became a reference to all researcher who implemented HHM method.

An HMM based method for Arabic phoneme recognition has been proposed by ELOBEID1 et al. in[102] . In this work, each of the Arabic phonemes are modeled using a discrete HMM.

Viterbi algorithms are implemented for training and recognition.

To decide the best features to represented Arabic speech signals, performance tests were implemented on several features representations such as prediction coefficients, area function, cepstral coefficients, etc.

the systems results showed the superiority of the cepstral coefficients, with a recognition performance of 74% over the other representations.

Results also showed that supplementing the cepstral coefficients with delta power and delta-delta power improves the performance to 81%.

They also present in [103] new parameters for resolving acoustic confusability between Arabic phonemes in a phonetic HMM recognition system.

Boreczky, J.S. and Wilcox ,L.D [104] introduced a technique for segmenting video using hidden Markov models.

They segmented a video into regions defined by shots, shots boundaries, and camera movement within shots.

The following features are extracted for segmentation phase: an image distance based on distance between adjacent video frames, an audio distance based on the acoustic difference in intervals just before and after the frames, and an estimate of motion between the two frames. Motion and audio information is used separately.

Their segmentation technique allows features to be combined within the HMM framework. Thresholds are not needed, because HMMs trained automatically.

They tested their algorithm on a video database, and has been shown to improve the accuracy of video segmentation over standard threshold-based systems.

HMMs can be used to classifying TV programs. For example Liu Zhu et al.[105] presented a system to recognized TV broadcast video based on HMMs.

The system implemented on five types of TV programs, these programs are: commercials, basketball games, football games, news reports, and weather forecasts. For each TV program, 20 minutes' videos from different TV channels are collected to construct the dataset. These datasets are divided into two sets. One set for training phase and other set for testing phase.

Features are extracted from low level audio (eight frame) and high-level audio (fourteen clip) properties. Ergodic HMM are used with different number of states (a ranged from 4 to 8 states). Further it implemented on different number of symbols. The overall accuracy rate is 84.7%.

There are many studies reviewed the handwritten text recognition (offline, online). They were applied on different languages such as Arabic, Latin, Chinese, Hindi etc.

HMM has been successfully applied in several large-scale datasets for offline Arabic handwritten.

It has advantages overall other recognition methods. The following are some of the works that were reported[106-108]. Due to thesis scope these studies limited to offline Arabic handwritten recognition systems only.

Dehghan, M., et al.[67] used a discrete HMM with right to left topology to design A holistic system for the recognition of handwritten Farsi/Arabic words.

The chain code directions histogram of the image has been extracted by a sliding window to represent the feature vectors. They constructed HMM for each word. Each model trained by Baum-Welch algorithm (60% of dataset).

To improve the recognition rate, the probability distributions of trained HMMs has been smoothing by SOFM codebook. They achieved better recognition rate in top10(69.47% before smoothing and 91.35% after smoothing when smoothing factor equal 0.001).

Märgner, V., et al. [61] presented a semi continuous one-dimensional HMM system. The system designed to recognize Arabic handwritten words (26459 words from IFN/ENIT).

The system consists of three steps. In the first step, skew, height, length, and baseline were normalized; baseline is normalized by using projection methods.

Features are extracted by sliding window with three columns in the second step.

On the final step, a semi continuous HMM models were constructed per character. Each model has seven states and three transitions.

Standard Viterbi Algorithm is implemented for training and recognition. The overall recognition rate reached to 89%.

Khorsheed, M.S. [109] tried to overcome the overlapping problem in cursive Arabic script by designing a single HMM.

This model consists of multiple models where each one built for representing one character. For example, we need four models to represent one model for the word"محمد" .

The model depends on structural features extracted from the manuscript words. Features are extracted after implementing Zhang Suen thinning algorithm. The feature vectors are converted to discrete symbols by Vector Quantization algorithm.

As mentioned previously, the system is single global model. It is created from ergodic character models. Each path through the global model represents a sequence of character which represents the desired word.

The system tested on sample consists of 405 Arabic manuscript characters and achieved 97% (in top 5) accuracy rate.

Benouareth, A., et al.[48] proposed HMM system to recognize offline Arabic words.The system implemented on IFN/ENIT benchmark database. It based on discrete HMM with explicit state duration.

After doing some preprocessing operations (baseline detection and thinning), they extracted the features by performing special segmentation methods to segment the word into frames.

Then, feature vectors are constructed, and they were combination between statistical and structural features (41 feature).

Vector quantization algorithm is used to map the continuous features vector to discrete features. Right to left HMM topology with three transitions is used to recognize the character (one HMM for each word character).

Because they used explicit state duration, the states of HMM are varied according to character length.

This technique is useful and increases the recognition rate. The models were trained and tested by standard Viterbi algorithm.

Results showed HMMs with explicit state duration (with Gamma distribution) improved the accuracy rate.

El-Hajj, R., et al.[110] attempted to solve the overlapping problem in Arabic character.

They assume that some characters have ascending and descending strokes. These strokes may be overlapped with two neighboring characters.

They improved their previous system[111] by adding extra HMM models to represent the contextual character models. Therefore, the HMMs are increased. Thus, the system has one dimension HMM for each Arabic character and contextual character models.

They depend on lower and upper baselines features.

The system tested on IFN/ENIT dataset and found that the contextual character models improved the recognition rate about 0.6 %.

From the previous works in offline Arabic handwritten recognition it appears clearly the following notes:

- IFN/ENIT dataset are widely used in several systems.
- Discrete HMM is more suitable for character and words recognition, because it is simple to adapt.

- Left to Right HMM topology is best choice and allow three transitions between the states. It also can be reserved according to the feature extraction technique as seen in Figure (3.4).
- Explicit state duration is more accurate to represent the data, in order to represent different characters' length.
- Gamma distribution increase the recognition rate.
- HMM can be used to represents the basic Arabic character models in addition to other supplementary characters.
- Viterbi Algorithm is commonly used to training the HMM.
- Continuous features vectors are always mapping to discrete features by Vector quantization algorithm.

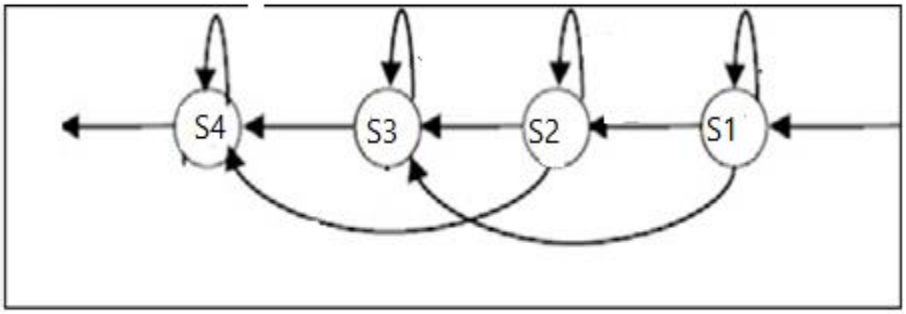


Figure (3.4): Right to Left HMM

As HMMs implemented in handwritten recognitions, it also implemented for electronic and handwritten signatures, for example Daramola, S.A. and Ibiyemi,T.S. [112] presented a recognition system for offline signatures using Discrete Cosine Transform (DCT) and HMM.

The signature to be trained or recognized is vertically divided into segments at the center of gravity using the space reference positions of the pixels.

Their recognition system is basically divided into five stages namely: data acquisition, preprocessing, feature extraction, training and recognition stages. The experimental result shows that successful signatures recognition rates of 99.2% is possible.

Another example for offline signature verification system was proposed by Justino, E.J., et al. [113]. The system based on HMM and graphometric features.

A dataset consist of 4,000 signatures samples from 100 authors are used for testing and validating the model.

Firstly a few a preprocessing process are done. Horizontal segmentation was used to divided written area into four zones with the same size (25 cells). The important parts of the text area are lower zone which describes the descenders part and the upper zone which describes the ascenders part.

In order to lowest the number of observation sequences. They used a 16-pixel segmentation to represents the signature width. Then static features were extracted (density of pixels in each cell and axial slant). Each column of cells is converted into a characteristic vector, where each vector element has a representative numeric value.

A discrete left-to-right HMMs model with two transitions were used in learning phase. Because each signature writer has a different signature size, therefore A HMM is constructed for each writer. Forward algorithm was used in the verification phase. The system achieved better results.

HMMs is also applied in protein domains.Terrapon, N., et al., [114] proposed system to fitting hidden Markov models of protein domains to a target species: application to plasmodium falciparum.

They used p falciparum as an example, they compared approaches that have been proposed for this problem, and presented two alternative methods. Their methods learn global correction rules that adjust amino-acid distributions associated with the match states of HMMs.

These rules are applied to all match states of the whole HMM library. Therefore, detection of domains from previously absent families is possible. They proposed a procedure to estimate the proportion of false positives among the newly discovered domains. They used Pfam standard library.

3.7 Conclusion

This chapter reviews the HMMs. HMMs may be continuous or discrete or hybrid. Discrete HMMs are more applicable. Three problems need to be solved in HMMs, these problems are: how to compute the probability of the observation sequence, how to find the optimal state sequence associated with a given observation sequence and how to find the appropriate method for adjusting the HMM parameters to maximize the probability of the observation sequence given the model. Several algorithms and methods are found to solve these problems.

We have shown the efficiency of the HMMs, it applicable in many applications especially in offline Arabic handwritten recognition.

CHAPTER FOUR

THESIS METHODOLOGY

4.1 Overview

We attempted to develop a system to recognized the isolated character body only.

It will be briefly that some Arabic characters have the same body, but differ only in the number and the position of their dots.

This similarity makes the recognition of isolated handwritten Arabic characters is a hard task. Therefore, in this thesis we take the characters' bodies and ignore the dots. Characters with similar body are grouped into one class, then each class represents by one character body, for example characters “Baa”, “Taa” and “Thaa” have the same class and represented by “Baa” body and so on for other characters.

SUST/ALT dataset is used to training and testing the proposed system. It consists of 47,988 isolated characters. But when dots are ignored the number of samples are decreased to 25179 characters.

Also, the number of the characters in dataset are decreased from 34 characters to 18 isolated Arabic characters.

The objectives of this chapter are to be discussed in detail the algorithms used in the preprocessing, feature extraction and classification phases. In this chapter detail descriptions of the methodology used in this thesis is introduced.

The chapter organized as follows: preprocessing phase introduced in Section (4.2) and the feature extraction phase described in Section (4.3). The classification phase by using HMM are discussed in Section (4.4). Finally, conclusions of the chapter summarize in Section (4.5).

4.2 Preprocessing Phase

Prior to the features extraction phase preprocessing phase must be done. Preprocessing of the handwritten character image is an important factor, to simplify the task of recognition[115].

Usually as mention in chapter 2 several operations cans be performed in this phase.

Since in SUST isolated characters dataset, some preprocessing method are done during the development stage[72],minimal number of preprocessing processes are used in this thesis.

An image file of isolated handwritten character will first be introduced to the system as gray scale bmp image. Then it read and normalized to fix size, which overcome the different style of the handwritten and maintain the devices. Then obtained images are binarized to be in digital form. Because the study focuses on characters' bodies only, dots are removed from some characters.

Thinning is very important process in OCR, therefore we applied it the binary images. The preprocessing operations are shown in Figure (4.1). The next sections give a brief detail of these operations.

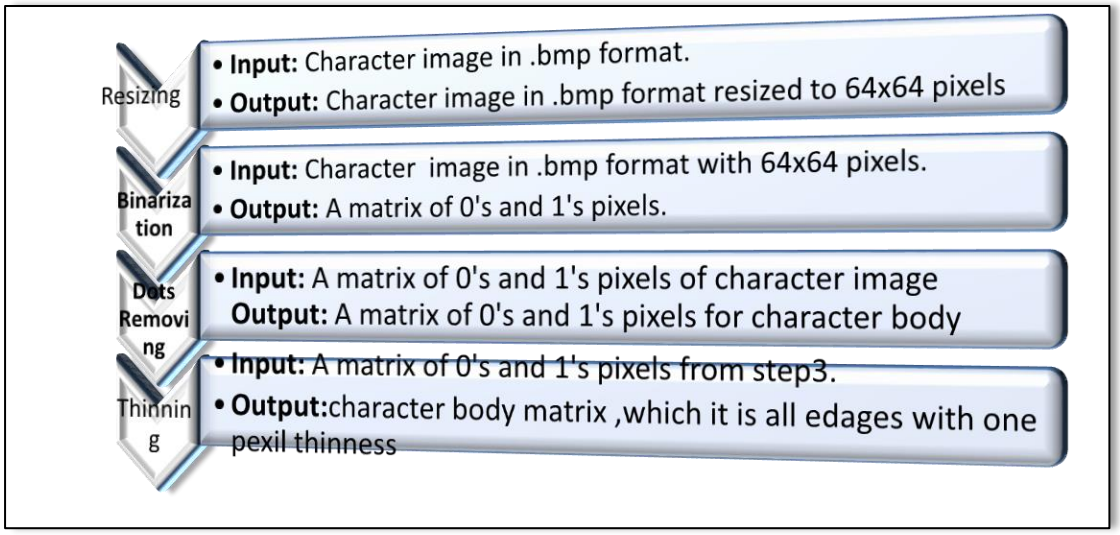


Figure (4.1): Preprocessing Operations.

4.2.1 Normalization

Normalization is aimed to adjust the size, position, and shape of character images.

Our purpose of this step is to reduce the shape variation of images of same class. Before reading the images, it normalized by resizing each image to a suitable size.

To choose the best size, different systems are studies[54, 116-118], 64x64 pixel founded to been the best one.

4.2.2 Binarization

Binarization operation attempts to convert the gray scale image into a binary image based on threshold. So, the bitmap images are threshold and converted into 1s and 0s forms.

As discussed in chapter 2 two types of thresholding are exist. These types are global and local thresholding.

In global thresholding, threshold selection leads to a single threshold value for the entire image[119].This value is often based on an estimation of the background intensity level of the using an intensity histogram.

In local thresholding different values are used for each pixel according to the local area information[31].

Since the proposed system implemented on simple isolated handwritten character images, where the characters can be to distinguish into background and foreground pixels, the global thresholding methods are sufficient for this type of images. Therefore we use Otsu's method[120], which used and adaptive by several researchers[121],[122].

It applied on dataset to convert the image into 1s (background) and 0s (foreground) pixels.

4.2.3 Dots Removing

The thesis goal here is to design an offline handwritten recognition system to deal with isolated Arabic handwritten character body written by multiple writers.

Before recognition all dots need to be removed. Some of Arabic characters, may have three, two or one dots such as " ث، ت،ب" characters or may be without any dot such as "ح ، د ، و" characters.

In this thesis, all dots are removed to obtain the character body only, Figure (4.2) shows sample of all characters used in this thesis after removing the dots

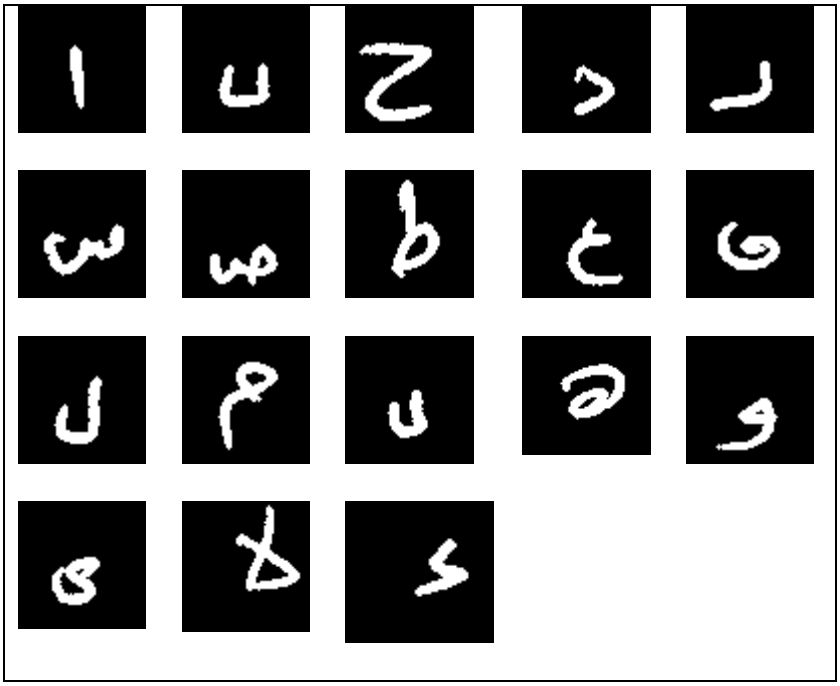


Figure (4.2): Sample of all Characters Bodies used in This thesis

4.2.4 Thinning

To improve the accuracy rate thinning operation are done. Various thinning algorithms are available.

The two importance constraints in thinning algorithms are:

- The end of the thinned image branches must not be shortened.
- Removal of border pixels must not change the skeleton of the image.

In this study, the thinning method presented in [123] has been used. It is parallel Thinning Algorithm proposed by Zhang, T. and Suen,C.Y.

In this method, the boundary and corner points of the character are removed, only a skeleton of the character remains.

It is a very effective method, because it decreases computation time, maintained the connectivity of skeleton and satisfied the above constraints, it usually used by researchers[43, 123-125].

A binary image defined by a matrix of pixels. Because the algorithm is parallel all the image points can be processed simultaneously. Each image element connected with eight neighboring elements.

It consists of two sub-iterations. The first sub-iteration focuses on deleting the south-east boundary points and the north-west corner points. The second sub-iteration focus on deleting the north-west boundary points and the south-east corner points as shown in Figure (4.3).

P_9 (i-1,j-1)	P_2 (i-1,j)	P_3 (i-1,j+1)
P_8 (i,j-1)	P_1 (i, j)	P_4 (i,j+1)
P_7 (i+1,j-1)	P_6 (i+1,j)	P_5 (i+1,j+1)

Figure (4.3): Matrix Represents P1 with Eight Neighbors Pixels

All the sub-iterations must have satisfied certain conditions. Here we explain the conditions of first sub-iteration, these conditions are:

- Condition1: $2 \leq B(P_1) \leq 6$, Where $B(P_1)$ is the non-zero neighbors of P_1 , B(P1) = P2 + P3 + P4 + • • • + P8 + P9.
- Condition2: $A(P_1) = 1$ Where $A(P_1)$ is the number of 0,1 patterns in the ordered of $P_2, P_3, P_4, \ldots, P_8, P_9$ that are the eight neighbors of P_1.
- Condition3: $P_2 * P_4 * P_6 = 0$.
- Condition4: $P_4 * P_6 * P_8 = 0$.

After applying all the conditions of first sub-iteration to all pixels of the image, the flagged pixels will be removed from the image.

Then the second sub-iteration is applied to the image.

The conditions of this sub-iteration are the same as conditions of first sub-iteration. It differs only in neighbors' pixels. These conditions are:

- Condition1: $2 \leq B(P_1) \leq 6$
- (the same condition as first sub-iteration)
- Condition2: $A(P_1) = 1$
- (the same condition as first sub-iteration).
- Condition3: $P_2 * P_4 * P_8 = 0$
- Condition4: $P_2 * P_6 * P_8 = 0$

The processes of Applying first sub-iteration and second sub-iteration are continued one after the other. It stopped when there are no more changes will occur in the image. When the processing finished, the image will be the thinning, our experimentation thinning result will be discussed in chapter 5.

Due to the similar shape of Arabic characters, researchers in Arabic handwriting face a problem after applying thinning algorithms ,[126].

To overcome this problem, we removed dots before applying thinning algorithms.

4.3 Feature Extraction Phase

The next step is to extract the useful features that will have used in classification phase. Many researchers[118],[127-128] agree that feature extraction phase play an important role in a handwriting recognition systems.

A human being can differentiate between various objects by observing their colors, shapes and attributes. To simulate this intelligence idea into a computer system, we need to implement geometrical and topological representation methods to help the system in recognizing the shapes of objects.

After studying several features methods, we found that geometrical and topological representation methods are reverent methods to recognize the isolated handwritten character.

To represent each image in a feature vector form, a mathematical model with a finite number of parameters is required. But unfortunately, they are no reasonable mathematical model currently exists.

As a result, there is no universally accepted set of feature vectors in document image understanding[28].

In this chapter, we describe an efficient method for extracting features from handwritten Arabic character body using freeman chain code as shown in Figure (4.4). The topological feature is the most useful to recognize handwritten image[129]. Ten features are used in this work, which obtained from normalized chain code.

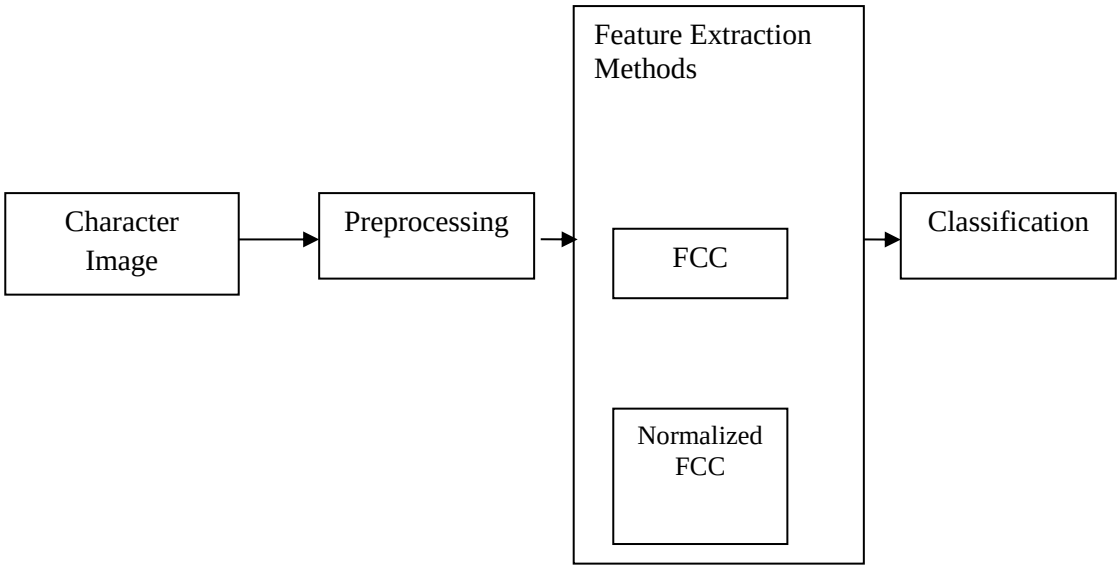


Figure (4.4): Feature Extraction Methods used in this Thesis

4.3.1 Freeman Chain Code Algorithm

There are several reasons effects our choice of FCC, these reasons are:

- It work well in sold connective object[130], and our dataset satisfied this condition because it represent the class body only.
- It is a simple algorithm.
- It passes through an image of a handwritten character as naturally as it was written.
- It an efficiency algorithm in term of storage utilization, especially when a large image is used[25].
- It facilitates the process of feature extraction.
- It preserves the information

FCC can be 4-connectivity or 8-connectivity. It traced the boundary of an image in a clockwise or anticlockwise directions. The main weakness of 4-connectivity is that we be unable to find the transverse points[131]. These points are very valuable in image recognition. So, to overcome the weakness of 4-connectivity we use 8-connectivity FCC. In 8-connectivity

each code can be considered as the angular directions (8 directions).

In our experimentation, we labeled the direction from 0 to 7 and decide to trace the image in a clockwise direction, it is commonly in many research and suitable for Arabic characters, because it written from right to left [132-134].

The big challenge of FCC is how to find the starting pixel and the directions of image traverses. We have produced different chain code for the same character, if we start with different starting points or traverse on different directions. Therefore, consistency plays an important factor to overcome the variations of the chain code for the same character and preserve the success of the algorithm.

As mention before, we obtained the boundary of the character image by traverse the FCC in a clockwise direction. We assigned numbers 0 to 7 to each direction.

The character images are binarized to 1's and 0's pixels in preprocessing phase. Instead of storing the absolute location of each 1 pixel, we stored it is direction from its previously coded neighbor. A neighbor is any of the adjacent pixels in the 3x 3 pixels' neighborhood surround that current pixel see Figure (4.5).

The literature of Freeman chain code was introduced in chapter two. To define the starting point, we move from the top of character body image to the bottom raw by raw to find first nonzero pixel. Furthermore, we assume that this first pixel has one neighbor. When we find the first nonzero pixel we defined it as starting point of the chain code and stored it in chain code list. In some cases, the first nonzero pixel has two neighbors. This case holds when the character body written as loop, for example, see the "Haa" character in Figure (4.6).

If this case arises, we assume the starting chain code is zero. After finding the start point of the chain code we traverse to the next neighbor pixel in the image of character body.

In fact, there must be at least one neighbor boundary pixel at one of the eight locations surrounding the current starting boundary pixel (note each location has one value from 0 to 7 marked per the chain code direction). Again, the starting point can have more than one neighbor. In this case the chain code direction plays and important factor, to determine which neighbor will be chosen. In our experimentation, we chose a clockwise direction. When the neighbor found, we stored it in chain code list. Then finding the next neighbor.

The process of finding the next neighbor continuous until we reached to the starting point. This algorithm followed to find the FCC are summarized in Figure (4.7).

5	6	7
4	Starting Pixel	0
3	2	1

Figure (4.5): Neighbors of Starting Pixel and Value Assign to them

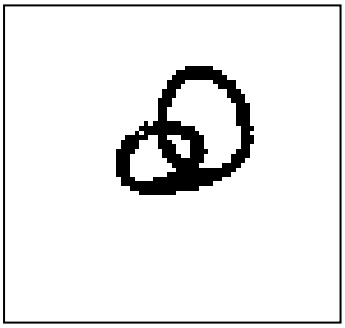


Figure (4.6): Haa Character Written as Loop

Input: Thinning binary character body image.

Output: 8-connective Freeman chain code

While there are still images to be traversed

Begin

move from the top of character body image to the bottom, raw by raw to find first nonzero pixel

If one nonzero pixel then

Begin

Assume that it is a starting pixel

Stored it is direction in chain code list

End

Else

Stored in chain code list 0 value as starting pixel

End

From starting pixel to end pixel do

Assign 0-7 values to the eight directions

Travels the neighbor pixel in clockwise direction

Find and store the direction code of the neighbor pixel in chain code list

Move to next position

End

End

Figure (4.7): The Algorithm for Generating 8- Connective Freeman Chain Code

4.3.2 Normalized Freeman Chain Code

The objective of this sub phase is to convert the one-dimensional matrix of chain code to two-dimensional matrix. Where first dimension of the matrix represents the value of the chain code, and the second dimension represents the frequency of that value. For example, if the chain code for character body is: 655577255554777740041443 4 it can be converted into two dimensions as following:

6	5	7	2	0	1	4	3
1	7	6	1	2	1	6	1

Then we remove some frequency from the chain code. Firstly, all values which their frequency is 1 are removed. In the chain code in the example 6, 2 , 1,3 values are removed, so the chain code decreased to:

5	7	0	4
7	6	2	6

The process for removing the value of less frequency will continued until we removed all values that their frequency is less than four. Then we used the following equation to normalize the chain code matrix to chain code with length of 10 digits:

$$F_i^n = \frac{F_i}{\sum F_i} \times 10$$

Where F_i^n is the normalized frequency and F_i is the frequency of each value in the chain code correspondingly. In the above example, the chain code matrix converted to:

5	7	4
3.68	3.16	3.16

then rounded the frequency to a near number and concatenated to

generate the length=10, in our example the final chain code will be:

5555777444

In some special case, the normalized chain code consist of less than 10 digits, when it found we repeat the last digits to reach 10 digits. The algorithm of normalized Freeman chain code illustrated in Figure (4.8)

Input: 8-connective Freeman chain code.

Output: Normalized Freeman chain code with 10 digits

1. *Converting Freeman chain code from one dimension matrix to a two dimensions' matrix. The first row for the chain code value and the second row for the frequency of that value.*
2. *Removing all values whose frequencies are less than four*
3. *Normalized frequencies to obtaining the final chain code with length of 10 by this equation:*

   $F_i^n = \frac{F_i}{\sum F_i} \times 10$ *Where* F_i^n *is the normalized frequency and* F_i *is the frequency of each value in the chain code*
4. *Rounded the frequency to a near number and concatenated to generate the length=10*
5. *Check whether the normalized chain code consist of 10 digits, if not repeat the last digits to reach10 digits*

Figure (4.8): The Algorithm for Normalized Freeman Chain Code

4.4 Classification Phase

Classification in OCR systems is the main decision making phase, because features extracted from the pattern are compared to features extracted from the model set[135].

There are several classifying methods for text recognition, among these HMMs is widely used[136-137].HMMs have been successfully applied to text recognition. Recognition of offline handwritten characters are the most researched topics involving HMMs[10],[138]. As seen in chapter 3, there are three main problems need to solve when using HMMs:

1 The evaluation problem: computing the probability that the observed sequence P(O/λ) was produced by the given model $\lambda = (A, B, \pi)$
2 The optimal state sequence problem: Finding the optimal sequence of hidden states, in a given model $\lambda = (A, B, \pi)$,that produced a given sequence of observations.
3 The training problem: how to find the method to adjust the parameters (A,B,π) to maximize the probability of the observation sequence.

In general, the HMMs consists of three main parameters called: the transition matrix probabilities, the emission matrix probabilities and the initial states probabilities.

There are many different topologies model for HMMs. In this thesis, a left to right HMMs topology is implemented for isolated Arabic handwriting characters' bodies recognition. This model allows the transition to the same state, the next state and to the following states only. HMMs are used in this thesis for the following reasons:

- There is no need to segmentation. we recognized the Arabic characters' body as whole unit.
- HMMs are easy to construct and validated.
- HMMs can be used as a classifier and construct modeling of the data at the same time.
- HMMs are resistant to noise, which able to deal with different writing style.

To develop a character's body recognition system based on the HMMs, the following procedures must be followed:

1. Choose the number of observation.
2. Choose the number of states.
3. Choose the HMM topology.
4. Select randomly the training and the testing samples.
5. Train the system using the training dataset.
6. Test the system using the testing dataset.

For isolated handwritten character recognition, it is useful to think about left to right topology[108, 135, 139]. In the left to right model transition from state i to state j is only permitted if $j \geq i$, smaller number of transition are occurred. Probabilities need to be learned (three transitions only).

The clusters of observation are formed for each model separately by estimating the Gaussian mixture parameter for each model. Therefore, left to right HMMs with three transitions are used in this thesis Figure (4.9) shows left to right model with six states.

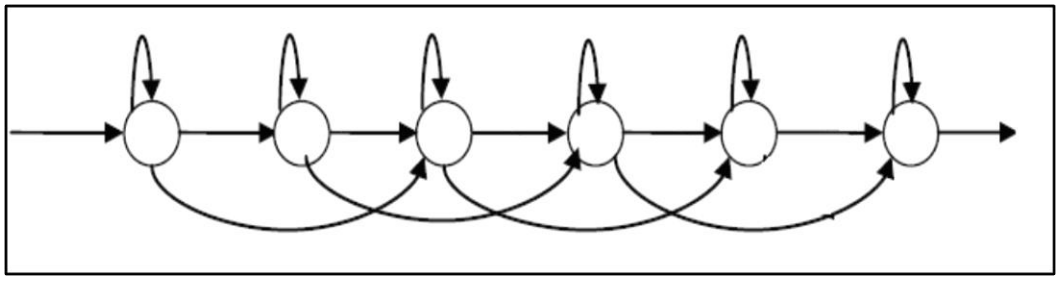


Figure (4.9): A Left to Right Model with Six States.

To recognized the character body image, the dataset is divided randomly into two sets. The first set consist of 70% samples from dataset, it used for training the models, while the remainder 30% samples are used for testing the models.

Baum-Welch and Viterbi Algorithms are used to develop the training and the recognition phases. These phases are described in detail in the following subsections. Figure (4.10) illustrated the training and testing Phases in the HMM Classifier.

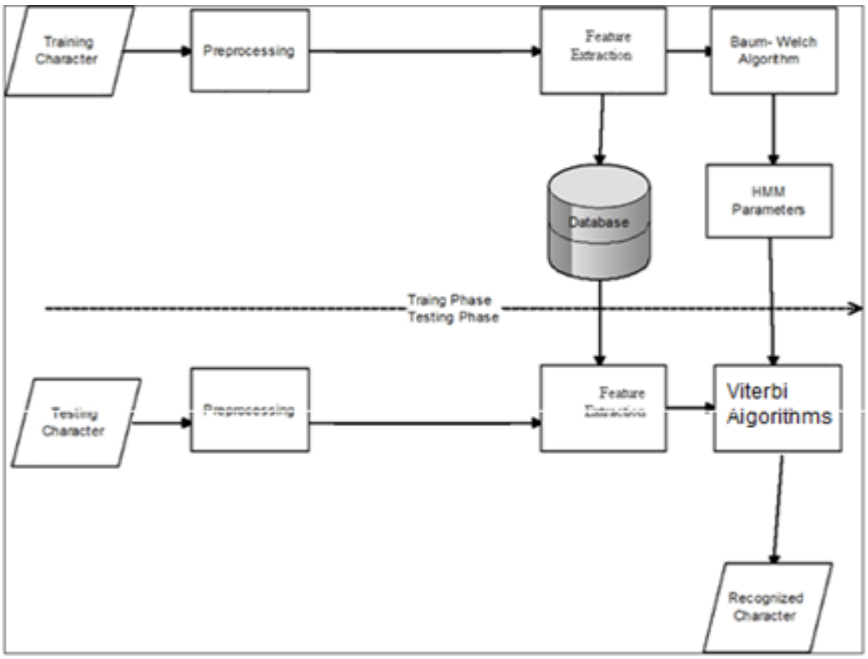


Figure (4.10): Training and Testing Phases for HMM Classifier.

4.4.1 Training Phase

In the training stage, the model is optimized by applying an iterative algorithm called the Baum-Welch algorithm to maximize the observation sequence probability $P(O|\lambda)$for the given model $\lambda = (A, B, \pi)$,to finding the Maximum Likelihood(ML).

In general, HMMs can be trained by the Baum-Welch Algorithm to give an acceptable performance[140].

By using the notation of HMMs elements as we describe in chapter 3

(Rabiner[118] notation),the following variables must be defined to obtained the Baum-Welch Algorithm:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda) \qquad \{4.1\}$$

This variable represents the joint probability of the partial observation sequence up to time *t* and the hidden state S_i at time *t* given λ.

$$\beta_t(i) = P(o_{t+1} o_{t+2} \cdots o_T \mid q_T = S_i, \lambda) \qquad \{4.2\}$$

which is represents the probability of the partial observation sequence from time *t*+1 up to *T* given λ and that the hidden state at time *t* is S_i.

Then the variable $\xi_t(i, j)$ is defined to represents the probability the hidden state at time *t* is S_i and at time *t*+1 is S_j given the observation sequence and λ.

$$\xi_t(i, j) = P(q_T = S_i, q_{T+1} = S_j \mid o_1 o_2 \cdots o_T, \lambda) \qquad \{4.3\}$$

$$= \frac{P(q_T = S_i, q_{T+1} = S_j, o_1 \cdots o_T)}{P(o_1 o_2 \cdots o_T \mid \lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(o_1 o_2 \cdots o_T \mid \lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}$$

which is represents the probability that the hidden state at time *t* is S_i and at time *t*+1 is S_j given the observation sequence and λ.

$$\gamma_t(i) = P(q_T = S_i \mid o_1 o_2 \cdots o_T, \lambda) \qquad \{4.4\}$$

$$= \frac{P(q_T = S_i, o_1 \cdots o_T)}{P(o_1 o_2 \cdots o_T \mid \lambda)}$$

$$= \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}$$

$$= \sum_{j=1}^{m} \xi_t(i, j)$$

which is represents the probability that the hidden state at time *t* is S_i given the observation sequence and λ.

Using these definitions, the HMM parameters $\lambda = (A, B, \pi)$ can be computed by the following procedure:

1. Initialize the parameters $\lambda = (A, B, \pi)$ randomly: initialize a_{ij} to *1/N*, b_{mp} to *1/M*, and π_i to *1/N*.
2. Follow the equations [4.1]to [4.4], to compute the parameters *αt(i), βt(i), ξt(i,j)* and *γt(m)*.
3. Compute the new parameters of the model $\lambda^* = (A^*, B^*, \pi^*)$ according to the values in step 2 as follows:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_t(i, j)}{\sum_{t=1}^{T} \gamma_t(i)} \qquad \{4.5\}$$

$$\bar{b}_{ij} = \frac{\sum_{t=1, O_t = v_k}^{T} \gamma_t(j))}{\sum_{t=1}^{T} \gamma_t(j)} \qquad \{4.6\}$$

$$\bar{\pi}_i = \gamma_t(i) \qquad \{4.7\}$$

4. The current values of parameters in step 2 and step3 are evaluated, then they are used to recomputed the parameters A, B, π iteratively.

4.4.2 Testing Phase

In the testing(Recognition)phase, the Viterbi Algorithm (VA) [91] is used for recognition purpose.

This algorithm is used to find a single best state sequence by applying the following steps[91]

Initialization step:

$$\delta_1(i) = \pi_i b_i(O_1)\,, \quad 1 \le i \le N \qquad \{4.8\}$$

$$\varphi_1(i) = 1 \qquad \{4.9\}$$

Recursion step:

$$For\ 2 \le t \le T\,, \quad 1 \le j \le N$$

$$\delta_t(i) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij} b_j(O_t)\right] \qquad \{4.10\}$$

$$\varphi_t(j) = \arg\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij} \qquad \{4.11\}$$

Termination step:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \qquad \{4.12\}$$

$$i_T^* = \arg\max_{1 \le i \le N} \delta_T(i) \qquad \{4.13\}$$

Path backtracking step:

$$For\ t = T-1\,,\ T-2, \ldots, 1$$

$$i_T^* = \varphi_{t+1}(j) i_{t+2}^* \qquad \{4.14\}$$

4.5 Conclusion

In this chapter, the proposed system is discussed. The system built for recognition of handwritten isolated Arabic characters' bodies based on HMM as classifier.

The phases of the system can be divided into three phases. These phases are: preprocessing, feature extraction and classification phases.

First the isolated handwritten character will be introduced to the system as bitmap image with bmp extension and the system read it. The dataset was divided into two parts, one part for training the system and the second part for testing the system.

After reading the image the preprocessing operations are done to the image. Firstly, the bmp image converted to digital image contain only 1’s for foreground and 0 ‘s for background. Then the width and high are normalized to 64x64 pixels. Because the study focuses only on the characters' bodies, dots are removable from some characters. Then the obtained digital image is thinning.

After preprocessing phase features are extracted from thinning images. FCC is used for this purpose. We used 8 connection type. To minimized the obtained FCC, we normalized it to ten bits only. The final phase of the proposal system is classification, which achieved by HMM. We built 18 left to right HMM for all the characters in our dataset.

CHAPTER FIVE

EXPERIMENTAL RESULTS AND DISCUSSIONS

5.1 Overview

This chapter discussed in detail the experimental results in each phase. The thesis consists of three main phases as illustrated in Figure (5.1).

The first phase was preprocessing phase, the second was feature extraction phase, and the third was classification.

Also in this chapter, the dataset used in this thesis are described, as well as experimental tools and software involved in each phase.

The chapter is organized as follows: dataset is introduced in Section (5.2). Experimental tools and software are introduced in Section (5.3) and (5.4). The preprocessing results are discussed in Section (5.5). Then the experimental results for feature extraction phase are introduced in Section (5.6). The practical implementation and evaluation of isolated characters' bodies recognition using HMMs classifiers are discussed in Section (5.7). Finally, Section (5.8) draws some conclusions from the work presented in this chapter.

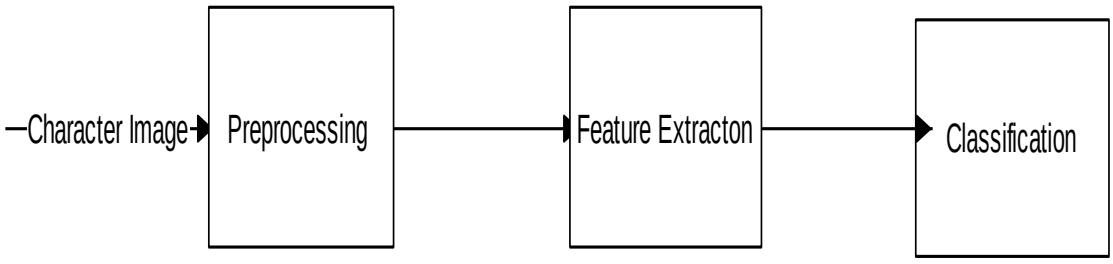


Figure (5.I): Phases of this Thesis

5.2 Thesis Dataset

Today, the researcher's trends to train and test their recognition systems on a large amount of real dataset. Especially, when characters forms are used.

One of our contribution is using large dataset. But how to find the suitable dataset? This is a big problem because the successful of the recognition depends on the dataset.

Therefore, we found SUST/ALT for isolated Arabic characters' dataset is appropriate one. We used it to train and test our proposed system for several reasons:

- It is a real dataset, that mean we avoid the disadvantage of using an artificial dataset.
- It is a large dataset contain more than 47,988 characters.
- It has truth information.
- It is clear dataset and free from noise, that limit the preprocessing phase to four operations only.
- It is written by different writers.
- It contains supplementary character Hamza(ء), it operates on character Alef(ا) to create Hamza on Alef and Hamza under Alef (أ,إ) characters. Furthermore, it operates on Waw character to create Hamza on Naberah and Hamza on Waw characters. In addition to these characters the dataset contain Lam Al Alef character(لا) is created as a combination of two characters, Lam (ل) and Alef (ا). These new characters, increase the number of Arabic characters from 28 characters to 34 characters.

From characteristics of Arabic language, it will be briefly that some characters have the same body, but differ only in the number and the position of their diacritics.

This similarity make the recognition of isolated handwritten Arabic characters is a hard task[141]. To facilitate the recognition phase, minimize the computation time and simplify the system, we develop a system to recognize the characters bodies only.

Therefore, in this thesis we take the character's bodies and ignore dots and diacritics. Characters with similar body are grouped into on class, then each class represents by one character body, for example character “Baa”, “Taa” and "Tha" have the same class and represented by “Baa” body and so on for other characters.

This scheme reduced the number of the characters in dataset from 34 characters to 18 characters' bodies. Also, number of samples are decreased to 25179 characters instead of 47,988 characters. So, the complete dataset consists of 25179, which is bigger than IFN/ENIT characters dataset. From this point of view the character recognition seems easier to achieve. The characters used in this thesis and their frequency are illustrated in Table (5.1). Figure (5.2) shows a set of 50 Samples of the isolated “Sad “character.

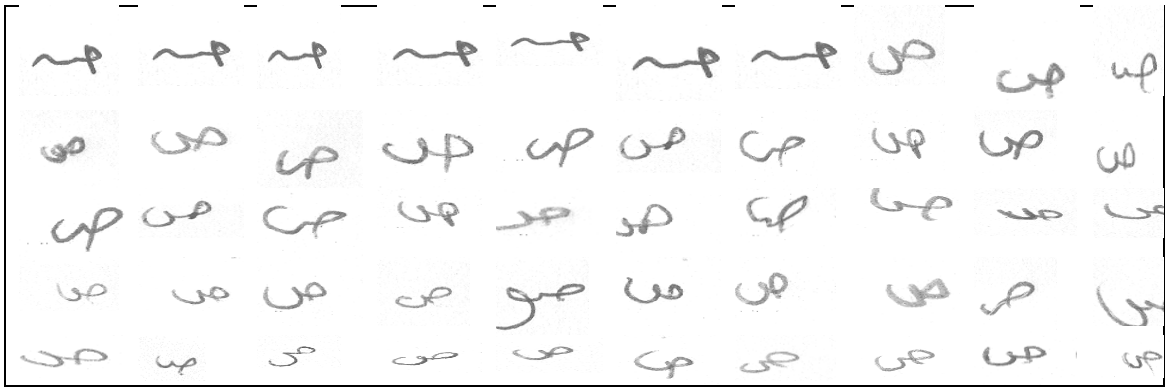


Figure (5.2): A Set of 50 Samples of the Isolated Sad "ص" Character from the Dataset.

Table (5.1): The Thesis Dataset

Class No	Character Body	Characters	Frequency
1		ا ، أ ، إ	1208
2		ب ، ت ، ث	1410
3		ج،ح،خ	1410
4		د ،ذ	1411
5		ر ، ز	1410
6		س ، ش	1410
7		ص ، ض	1410
8		ط ، ظ	1410
9		ع ، غ	1410
10		ف ، ق	1410
11		ك ,ل	1410
12		م	1410
13		ن	1410
14		هـ	1410
15		و ، ؤ	1410
16		ي ، ئ	1410
17		لا	1410
18		ء	1410
Total			25179 characters

5.3 Experimental Tools

In this experiment, the Laptop computer (HP) that was used with the following specifications: Microsoft Windows 10, home edition, 64 bits, year 2016, Intel processor Core(TM) i3,2.00 GHz with 4.00 GB RAM memory.

5.4 Software Used

MATLAB software (R2101b) was used for programming all the system phases (preprocessing, feature extraction and classification phase), because it offers several advantages as summarized below:

- It consists of a variety of tools.
- It provides mathematically based analysis.
- It easily to be developed, debug and maintain.
- It minimized the computation time.
- It provides graphical solutions to a problem.
- It can be used to analyze and solve problems, rather than a languages that used to produce executable applications[142-143]

Also, Hidden Markov Model (HMM) Toolbox written by Kevin Murphy, 1998. Last updated: 8 June 2005 [144]are used in classification phase.

It distributed under the MIT License. It supports inference and learning for HMMs with discrete outputs "DHMM", Gaussian outputs "GHMM", or mixtures of Gaussians output "MGHMM"'. It also supports discrete inputs.

It freely downloaded from the following web site.

http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm_download.html

5.5 Preprocessing Phase

The main advantages of preprocessing phase are to systematize the information and simplify recognition task.\In this phase we attempt to eliminate some of the variations in the character images that do not affect the identification of the character[13],[137].

The previous chapter described a preprocessing methodology, which applied to an isolated characters images.

In this phase the character image is loaded as bitmap image, then the images are normalized by reducing the image size from 88X88 pixels to 64X64 pixels in order to reduce the computation time[118],[145].

Figure (5.3) illustrates the adjust character "Seen" image size to 64x64 pixels.

Figure (5.3): The adjust Character "Seen" Image size to 64x64 pixels.

Then the normalized image is converted to binary image.

The output binary image has values of 1 for all white pixels and 0 for all black pixels[145-146].

Figure (5.4a) displays an image of the character "د" in .bmp file format, and Figure (5.4b) below displays the same character after it had been binarized and saved as text (.txt) file.

Because this thesis focuses on recognition the characters' bodies, dots are removed from the (Bah, Fah, Noon and Yah) characters images to extract the characters' bodies only.

Figure (5.4 a): An Image of Character "د" in bmp Form

Figure (5.4 b): An Image of Character "د" After Binarization and Saved in (.txt) file

Because this thesis focuses on recognition the characters' bodies, dots are removed from the (Bah, Fah, Noon and Yah) characters images to extract the characters' bodies only.

When applying this operation some of data are loss during this process). This method has been tested on 5640 characters' images and the results of this method is illustrated in Table (5.2) bellow.

Table (5.2): The Accuracy rate of Dots Removing

Character	**Accuracy Rate**
Baa	99.97%
Fah	99.50%
Noon	99.86%
Yaa	98.94%

Samples are loss due to two reasons. The first reason, is the writing style of the writers, for examples some writers connected dots with the main body of characters "Baa, Fah and Yaa", or writing dot inside character "Noon" as shown in Figure (5.5).

Figure (5.5 a): The Original Images

Figure (5.5 b): Images After Dots Removing

The second reason, due to unclear samples from the original dataset, Figure (5.6a) displays some unclear samples from the original dataset, and Figure (5.6b) displays the same samples after removing dots.

Figure (5.6a): Unclear Samples from the Original Dataset

Figure (5.6b): The same Samples after Removing Dots.

Finally, the character body image is thinned and edges are extracted which is be the input to the feature extraction phase.

Figure (5.7a) displays an image for character "Baa" body before thinning, and Figure (5.7b) displays the same image after thinning.

Figure (5.7 a) Image Before Thinning

Figure (5.7b) Image After Thinning

5.6 Feature Extraction Phase

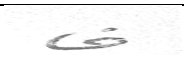
In this section, the feature vector for each character body image is obtained by applying 8-connective chain code. Then the two-dimensional (2D) matrix is converting to one-dimensional (1D) contain 10 digits by applying the proposed normalized chain code algorithm.

Results obtained from implementation these algorithms on character "ف" samples are shown in Table (5.3).

The proposed algorithm worked well and gives 99.23% result, this makes the algorithm more effective only about 193 of the samples their chain code be zero, this due to unconnected properties of these samples.

In making comparisons with the existing work, it is difficult to compare with work in[17] since those authors used other dataset and they have chosen 200 images so the proposed algorithm cannot be implemented on the same data.

Table (5.3): Results Obtained from Implementation of Normalized Chain Code on Character "ف"

No	Original Image	Image After Preprocessing Operations	Normalized Chain Code
1			5 5 6 6 7 0 0 1 2 3
2			5 5 6 7 7 0 0 1 2 3
3			5 5 6 7 1 1 0 2 3 4
4			5 6 7 0 0 1 2 3 4 4
5			5 5 7 6 7 0 0 1 2 3

5.7 Classification Phase

A left to right discrete hidden Markov models (DHMMs) was used to classify Arabic characters' bodies.

The system described in this thesis has been applied to SUST /ALT of isolated offline Arabic handwritten characters.

Samples of about 24857 characters are used. About 1.30% of the data was deleted from the testing and training data because of errors in preprocessing and feature extraction phases as described in previous section.

Hidden Markov Model (HMM) Toolbox written by Kevin Murphy, 2005 are used to classify the characters' bodies.

To classify a sequence into one of 18 classes, we trained 18 HMMs, one per class, and then compute the log-likelihood that each model gives to the test sequence; if the i'th model has the biggest log likelihood, then declare the class of the sequence to be class i. In other words, the character associated with the HMM with the highest log-likelihood was declared to be the recognized character.

In these experiments, each time 70% of the samples in the dataset were used for training and the remaining 30% for testing the system.

Many experiments have been conducted to find best values of recognition rate (mean).

We trained and tested the system with 18 classes set for classes described in Table (5.1). To decide the best number of state that given high recognition rate we used 3,5,7 states. The training and testing recognition rate of each character body class with different states (3,5,7) are illustrated in Table (5.4).

Table (5.4): Train and Test Recognition Rates for Characters with different Numbers of States

No	Classes	Recognition Rate%					
		3 States		5 States		7 States	
		Testing	Training	Testing	Training	Testing	Training
1	Alef	83.61%	93.56%	86.39%	94.04%	86.11%	93.80%
2	Baa	35.52%	57.72%	39.42%	59.92%	37.47%	58.14%
3	Hah	51.30%	73.88%	48.94%	73.58%	52.48%	73.98%
4	Dal	59.81%	87.06%	62.92%	89.01%	60.05%	88.30%
5	Raa	63.16%	85.03%	63.88%	89.23%	67.22%	90.15%
6	Seen	43.47%	79.12%	44.89%	79.43%	42.28%	76.27%
7	Sad	45.84%	77.88%	48.69%	78.08%	51.07%	78.49%
8	Tah	44.18%	77.29%	46.08%	76.99%	45.61%	77.50%
9	Ain	59.05%	84.49%	60.48%	85.31%	62.38%	86.42%
10	Faa	46.56%	75.46%	48.22%	78.11%	48.69%	79.84%
11	Lam	46.65%	77.54%	51.67%	78.05%	50.24%	79.08%
12	Meem	55.34%	81.47%	62.47%	83.91%	62.00%	81.57%
13	Noon	43.03%	72.52%	43.97%	72.72%	43.50%	71.91%
14	Heh	61.95%	83.33%	60.73%	83.23%	62.93%	83.36%
15	Waw	62.68%	84.60%	64.59%	85.73%	59.57%	83.16%
16	Yaa	51.42%	76.30%	51.90%	76.50%	48.82%	75.48%
17	Lam Elalef	58.47%	85.96%	61.58%	88.83%	60.38%	88.22%
18	Hamza	57.11%	84.28%	59.85%	82.89%	57.36%	84.39%
Average		53.55%	79.86%	55.93%	80.86%	55.45%	80.56%

The above table shows the highest recognition rate for testing dataset is 86.39% occurred with 5 states and 360 samples from Alef class as inputs. The recognition rates for some characters are very low, this due to the variation of writers' styles which case similarity between characters. For example, the highest testing recognition rate of character "ب" is 39.42%. Moreover, we find 5 states give best recognition rate for testing and training sets.

Also, most of the characters' bodies have best recognition rates with 5 and 7 states except "هـ" character because it is short character.

Figure (5.8) shows the relation between testing and training recognition rate according to the number of states.

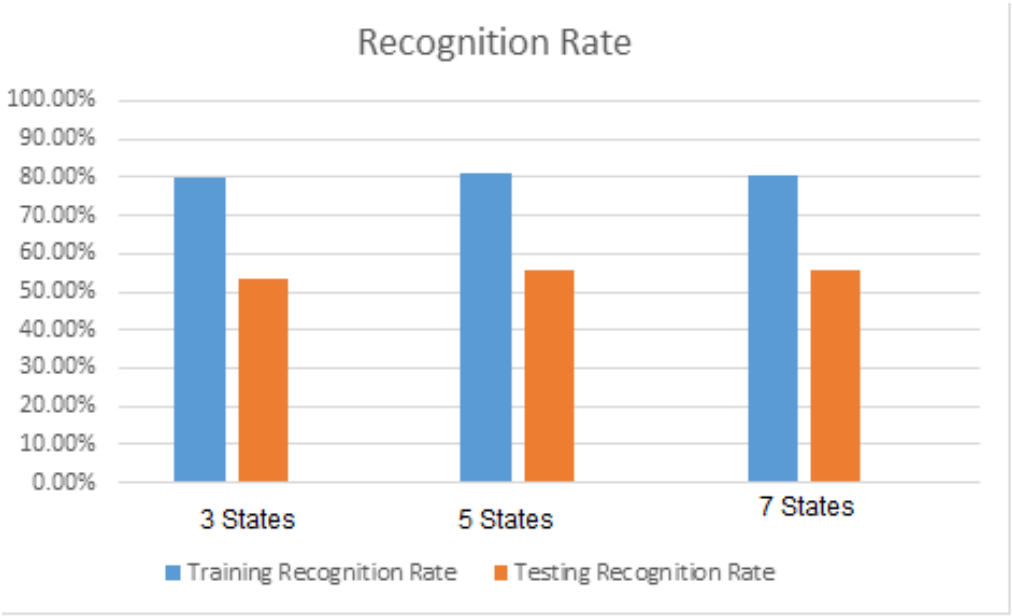


Figure (5.8): Testing and Training Recognition Rate

Table (5.5): Three States Confusion Matrix

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
1	310	0	0	1	0	7	8	8	1	0	4	4	0	2	1	0	12	2	360
2	0	146	11	20	21	23	12	6	3	20	20	13	77	5	10	10	11	3	411
3	4	1	217	18	7	2	5	14	25	13	1	28	11	19	19	13	12	14	423
4	2	2	17	250	40	5	7	4	6	16	5	7	4	4	17	3	5	24	418
5	2	2	8	15	264	14	8	18	2	15	7	15	10	5	20	1	10	2	418
6	2	6	11	17	20	183	31	14	0	23	11	21	15	8	16	14	16	13	421
7	4	5	9	6	8	7	193	25	12	16	2	28	12	22	15	22	16	19	421
8	9	13	11	14	18	16	9	186	9	20	9	10	2	24	13	19	29	10	421
9	6	0	1	4	3	2	13	4	248	9	9	77	2	8	4	20	3	7	420
10	4	9	6	11	18	9	23	22	6	196	6	15	12	13	26	19	7	19	421
11	14	20	9	18	24	17	18	17	5	15	195	10	26	3	6	3	17	1	418
12	7	0	14	6	5	12	15	26	20	22	1	233	1	18	9	7	14	11	421
13	1	30	21	24	24	21	15	9	10	13	10	21	182	9	9	6	11	7	423
14	0	2	4	3	3	1	14	20	8	9	4	20	2	254	24	10	10	22	410
15	3	0	2	15	7	2	16	15	3	20	6	14	2	23	262	3	12	13	418
16	3	3	5	2	1	1	27	16	10	19	3	37	5	17	20	217	7	29	422
17	3	5	11	14	11	5	2	31	3	12	20	8	15	10	11	12	245	1	419
18	1	1	0	20	8	7	8	16	22	16	3	17	1	16	18	17	1	229	401

Table (5.6): Five States Confusion Matrix

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
1	311	0	0	3	0	8	7	8	1	0	4	3	0	2	1	0	9	3	360
2	1	162	4	20	22	25	8	0	2	23	21	16	69	4	12	12	7	3	411
3	2	2	207	23	10	0	3	15	23	16	5	38	9	16	22	0	25	7	423
4	3	2	16	263	39	4	6	3	8	13	0	9	2	5	17	2	4	22	418
5	2	6	6	12	267	18	7	12	4	12	6	13	10	9	20	3	8	3	418
6	4	10	13	16	20	189	19	14	0	10	14	34	20	7	11	12	19	9	421
7	3	3	6	4	6	14	205	26	11	23	5	32	9	16	19	5	17	17	421
8	9	4	13	11	10	16	15	194	9	10	14	16	3	13	26	16	30	12	421
9	2	0	2	3	2	1	3	2	254	8	13	87	2	13	3	17	3	5	420
10	4	14	4	13	18	10	27	19	7	203	5	12	4	18	25	12	8	18	421
11	5	8	17	24	40	19	23	19	5	1	216	1	24	6	2	0	8	0	418
12	6	0	14	4	4	6	12	19	22	19	3	263	2	13	10	8	2	14	421
13	0	41	14	11	11	23	15	14	10	13	20	18	186	6	12	10	19	0	423
14	0	1	5	2	2	3	12	22	7	15	3	23	3	249	21	11	10	21	410
15	4	2	2	15	2	1	19	17	5	21	1	20	0	10	270	10	8	11	418
16	2	4	6	1	2	0	20	17	15	27	3	32	3	12	16	219	13	30	422
17	6	8	7	9	9	6	2	28	2	15	15	10	7	10	10	7	258	10	419
18	3	2	3	22	6	9	9	18	18	15	0	18	0	7	17	14	0	240	401

Table (5.7): Seven States Confusion Matrix

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Total
1	301	0	1	2	4	3	7	5	3	0	11	5	3	0	0	1	14	0	360
2	0	154	1	16	20	23	11	7	3	23	20	11	76	7	15	10	9	5	411
3	1	0	222	15	11	5	12	14	25	21	2	32	0	10	5	20	19	9	423
4	1	2	15	251	42	5	10	6	5	15	3	8	4	6	16	3	7	19	418
5	3	1	5	22	281	7	9	13	3	9	2	9	13	9	15	0	9	8	418
6	6	6	15	19	25	178	30	19	5	13	4	25	23	1	9	16	19	8	421
7	4	1	1	3	4	2	215	17	10	28	0	44	12	24	17	6	15	18	421
8	8	7	9	12	10	16	17	192	13	25	4	15	6	22	20	11	22	12	421
9	1	0	1	4	3	1	7	5	262	8	13	63	2	17	7	17	6	3	420
10	4	9	9	16	16	12	20	21	3	205	13	18	6	12	24	21	3	9	421
11	7	15	16	16	39	17	16	17	6	7	210	5	22	4	6	4	6	5	418
12	6	0	14	4	4	6	12	19	22	19	3	263	2	13	10	8	2	14	421
13	0	21	15	11	30	27	13	14	11	11	23	20	184	6	6	8	18	5	423
14	0	1	5	3	2	2	19	10	12	33	7	27	0	200	26	22	16	25	410
15	3	0	5	15	1	0	5	19	5	27	2	27	5	12	249	11	14	18	418
16	1	3	1	4	2	5	26	19	14	26	1	35	1	15	22	206	14	27	422
17	4	5	13	10	13	10	4	30	2	17	11	6	12	8	7	7	253	7	419
18	1	2	1	19	7	8	11	17	27	19	1	16	1	9	19	12	1	230	401

The following issues are observed from the above test confusion matrixes:

- Misclassification errors for many samples. These errors occur for several reasons, such as the samples being written by different writers using different styles. These errors may allow a character to have more than one model. Also, some samples are similar in shape. Moreover, error may occur due to inadequate capability with the features used.
- Alef class has few samples confused with other classes, because his shape is vertical and different from other classes which makes it in contrast with other classes in the chain code directions.
- Although some classes are near similar in the shape but they are not confused for example "Lamelef" and "Lam" characters similar in shape but their chain code is different because the loops in "Lamelef" character change chain code direction.

Table (5.9) below displays the classes that has high confusion with other

classes (more than 19 confusions is considered high confusion).

From this table, we find that five classes have high confusion, these classes are: Bah, Rah, Meem, Fah, and Waw. The rest of classes confuse with four classes or less except "Bah" and "noon" classes confuse with six classes in 3 states and with five classes in 5 and 7 states.

Writing style play the key role in this confusion Figure (5.9) displays samples of "ب" character recognized as "noon " character.

Also, Figure (5.10) shows examples of "د" character that recognized as "ر" character.

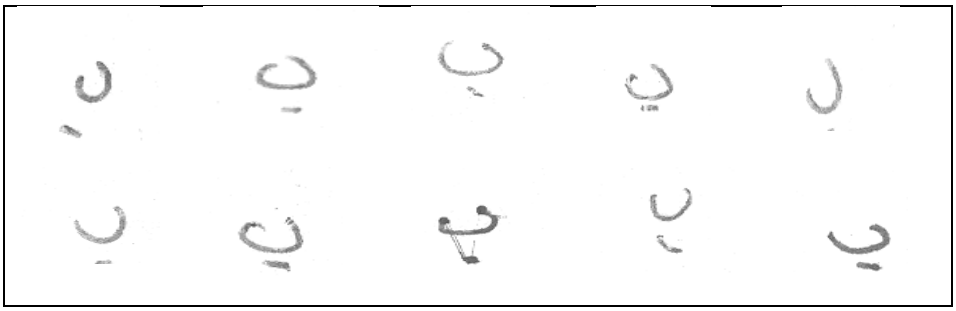


Figure (5.9): Samples of "ب" Character Recognized as "noon " Character

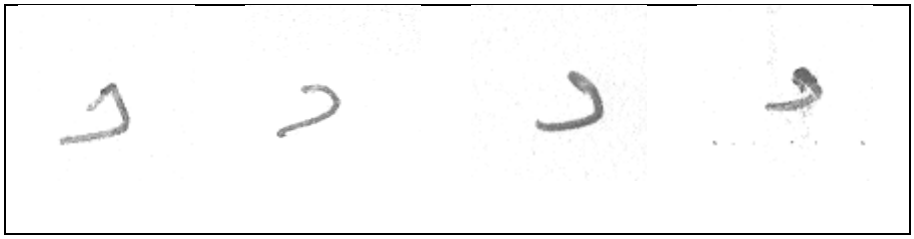


Figure (5.10): Samples of "د" Character Recognized as "ر " Character

Table (5.8): Classes that Confused with Other Classes

No	Classes	3 States	5 States	7 States
1	Alef (ا، أ ، إ)	_____	_____	_____
2	Baa (ب ، ت ، ث)	Dal, Raa, Seen, Faa, Lam, Noon	Dal, Raa, Seen, Faa, Noon	Raa, Seen, Faa, Lam, Noon
3	Hah (ج،ح،خ)	Meem, Ain	Meem, Dal, Ain, Waw	Meem, Ain
4	Dal (د ،ذ)	Raa, Hamza	Raa, Hamza	Raa
5	Raa (ر ، ز)	Waw	Waw	Dal
6	Seen (س ، ش)	Raa, Sad, Faa, Meem	Raa, Meem, Noon	Raa, Sad, Meem, Noon
7	Sad (ص ، ض)	Tah, Meem, Haa, Yaa	Tah, Meem, Faa,	Faa, Meem, Haa
8	Tah (ط ، ظ)	Faa, Haa, Lamelalef	Waw, Lamelalef	Faa, Haa, Waw, Lamelalef
9	Ain (ع ، غ)	Meem, Yaa	Meem	Meem
10	Faa (ف ، ق)	Sad, Tah, Waw	Sad, Waw	Sad, Tah, Waw, Yaa
11	Lam (ل, ك)	Baa, Raa, Noon	Raa, Sad, Noon	Raa, Noon
12	Meem (م)	Tah, Ain, Faa	Ain	Ain
13	Noon (ن)	Baa, Haa, Dal, Raa, Seen, Lam,	Baa, Seen, Lam,	Baa, Raa, Seen, Lam, Meem
14	Haa (هـ)	Tah, Meem, Waw, Hamza	Tah, Meem, Waw, Hamza	Faa, Meem, Waw, Yaa, Hamza
15	Waw (و ، ؤ)	Faa, Haa	Faa, Meem	Faa, Meem
16	Yaa (ي ، ئ)	Sad, Meem, Waw, Hamza	Sad, Meem, Faa, Hamza	Sad, Meem, Waw, Hamza
17	Lamelalef (لا)	Tah, Lam	Tah	Tah
18	Hamza (ء)	Dal, Ain	Dal	Ain

The proposal system gave low recognition rates. These low rates are due to the variation of writing styles of different writers.

One solution to improve the recognition rate is to decrease the number of groups and gather all similar characters into these group. We expect when grouped classes into few classes the recognition rate will increase.

Therefore, we gathered character "ن" into Baa class because most of Baa class samples were classified as Noon character, so the new character in "Baa" class are "ب ، ت ، ث،ن".

Also, misclassification occurred in six classes due to the confusion with "و" character. Therefore, we gathered it in "Faa" class because it's near similar to it. By this scheme the new classes be 16 class instead of 18 class that described in Table (5.1), because Noon and Faa classes were eliminated from Table (5.1).

Then the new grouped dataset is tested again with 16 class and five states because five states give the best recognition rate in the previous experiments, the results for the new experiments are shown in Table (5.10). As we show the testing recognition rate of all classes is 59.28% which is bigger than testing recognition rate of 18 classes 55.93%. That means grouping the similar samples into small set of groups will give better recognition rate.

More grouping of similar samples can be possible. For example researchers in [77] used the same dataset as us and grouped 34 isolated character into 15 and 11 classes. Unfortunately, we are unable to group characters like these, because of the future schemes of the spotting. For example, they group "ب، ت، ث، ن، ش" into one class. If we do this, how can we distinguish between"ث ،ش" characters in spotting? Both has three dots above. Therefore, we stopped grouping scheme into 16 class only.

In the future, we will follow schemes described in (Section 6.2) to increase the recognition more and more. The results achieved in this thesis encourage further researches in several areas.

Table (5.9): Test Recognition Rates for 16 Class using 5 States

No	**Class**	**Testing Recognition Rate**
1	Alef	86.39%
2	Baa	46.23%
3	Hah	50.59%
4	Dal	66.75%
5	Raa	67.22%
6	Seen	45.61%
7	Sad	50.36%
8	Tah	47.74%
9	Ain	61.68%
10	Faa	53.44%
11	Lam	54.55%
12	Meem	63.90%
13	Haa	61.46%
14	Yaa	66.59%
15	Lam elalef	64.88%
16	hamza	61.10%
Average		59.28%

5.9 Conclusion

In this chapter, the experimental details have been explained.

First the preprocessing methods were implemented, due to writing styles 196 samples from the dataset are failed when applying dots removing and thinning preprocessing methods.

Then FCC features are extracted from the thinned images also in this phase we deleted some samples from the dataset because their chain code are zero. Then FCC for each sample are normalized to 10 digits.

Finally, the performances of HMMS classifiers used to recognize handwritten are evaluated. Different models are designed (three, five and seven states) and four experiments were examined. Best testing recognition rate achieved when using five state models and 16 class (59.28%).

This result is low, due to several reasons. One of them is the writing style. Also, the features used was inefficient, more works are required in future that open opportunities for researchers.

CHAPTER SIX

CONCLUSION AND SUGGESTION FOR FUTURE WORK

6.1 Conclusion

In this thesis, an attempt has been made to design a system for offline isolated Arabic handwritten character recognition. The method is based on HMM to recognize isolated Arabic characters' bodies using Freeman chain code based on edge tracing. The thesis has focused on recognizing Arabic handwritten characters'. In particular, offline recognition of the Isolated Arabic handwritten characters is the main domain of this thesis.

The scope of the thesis is important for Arabic language, and other languages including Kurd, Farsi, Persian, and Urdu, because these groups use Arabic characters in their writing.

This thesis uses freeman chain code technique, which extracted the character body edges. The proposed method obtains the edges of the character body and then generates a freeman chain code for characters' bodies. The obtained Freeman chain code is normalized to 10 digits which passed as features to HMMs classifiers. It is believed that this work is important because HMMs recognition system design has been based on chain code features.

Since isolated Arabic handwriting recognition system depends on accurate preprocessing methods, preprocessing methods for Arabic characters are presented in thesis. These methods include method for normalization, binarization, dots removing, and thinning.

The dataset that used in this thesis is isolated Arabic character set developed by SUST/ALT group. It collected from 141 writers, each person

written each character ten times and scanned with 300 dpi. The total of all characters is 47,988 characters. In this thesis, we follow scheme to classify isolated Arabic characters' bodies without dots and diacritic, all similar characters are group into one class. Instead of having all 34 of the Arabic characters in the dataset, the system has 18 class, which represent Arabic characters' bodies after grouping similar characters into classes.

The total samples of 18 class (characters) is 25179 characters are divided into two sets a training set (70%), and testing set (30%). Features were extracted from images using freeman chain code (FCC). In the classification, discrete hidden Markov models (DHMM) classifier has been used to training and testing dataset.

In the first set of experiments the proposed classifier is trained and tested with 18 classes. This set of experiments have been repeated three times. In the first group, HMMs have three states. The second and the third groups have five and seven states respectively.

The Experimental results show that five states model gives the best recognition rate. So, the recognition rate for testing dataset was 55.93% and 80.86% for training dataset (using five states). One of the important finding of these set of experiment is the high confusion between " ن" and "ب"classes, and ، "و" and "ف" classes. Therefore, we design new grouping schema that contains 16 classes. As the first set of experiments show the best number of states is 5, only 5 states HMM were used in second experiments.

The recognition rate for testing dataset in the second experiment was 59.28%. This result is low, due to several reasons. One of them is the writing style.

Also, the features used was inefficient, more works are required in future to improvement the feature extraction method.

moreover, FCC feature is seemed to be not 100% relevant to use with HMMs, therefore other feature extraction need to be added to extracted useful features.

6.2 Suggestion for Future Work

The main objective of this thesis is the recognition of isolated Arabic handwritten characters' bodies. Acceptable accuracy, have been achieved. However, there is still many ideas look worth testing to improve this model. Suggestions for these future work ideas are below:

- More preprocessing operation can be added and examined, for example baseline estimation, noise detection and slant and slope correction may increase the recognition rate. Furthermore, the thinning algorithm presented in chapter 4 needs to be adaptive to improve the accuracy rate.
- Some characters like "لا, هـ" contains loops, algorithm to segment loops in characters may increase the recognition rate.
- Besides using the freeman chain code (FCC), other methods such as Fourier transform and Gabor filters might be added to extracted features from isolated characters' images. Adding different feature topologies may increase the overall system performance.
- Other classifiers such as K Nearest Neighbor (kNN), Support Vector Machines (SVMs) and recurrent neural networks can be used with our proposed grouping method and feature extraction method.

References

[1] AlKhateeb, J.H.Y., *Word based off-line handwritten Arabic classification and recognition. Design of automatic recognition system for large vocabulary offline handwritten Arabic words using machine learning approaches*, 2010, University of Bradford.

[2] Nasien, D., H. Haron, and S.S. Yuhaniz, *The heuristic extraction algorithms for freeman chain code of handwritten character*. International Journal of Experimental Algorithms-IJEA, 2011. 1(1): p. 1-20.

[3] Khedher, M. and G. Abandah. *Arabic character recognition using approximate stroke sequence*. in Proc. Workshop Arabic Language Resources and Evaluation: Status and Prospects and 3rd Int'l Conf. on Language Resources and Evaluation (LREC 2002*)*. 2002.

[4] Liu, J., J. Sun, and S. Wang, *Pattern recognition: An overview*. IJCSNS International Journal of Computer Science and Network Security, 2006. 6(6): p. 57-61.

[5] Watanabe, S., *Pattern recognition: human and mechanical*. 1985: John Wiley & Sons, Inc

[6] Mori, S., H. Nishida, and H. Yamada, *Optical character recognition*. 1999: John Wiley & Sons, Inc.

[7] Srihari, S.N., A. Shekhawat, and S.W. Lam, *Optical character recognition (OCR)*. 2003.

[8] Amin, A., *Off-line Arabic character recognition: the state of the art.* Pattern recognition, 1998. 31(5): p. 517-530.

[9] Cheriet, M., *Visual recognition of Arabic handwriting: challenges and new directions*, in *Arabic and Chinese Handwriting Recognition*. 2008, Springer. p. 1-21.

[10] Khorsheed, M.S., *Off-line Arabic character recognition–a review*. Pattern analysis & applications, 2002. 5(1): p. 31-45.

[11] Bortolozzi, F., et al. *Recent advances in handwriting recognition*. in *Proceedings of the International Workshop on Document Analysis*. 2005.

[12] Al-Taani, A.T., *An efficient feature extraction algorithm for the recognition of handwritten arabic digits.* International journal of computational intelligence, 2005. 2(2): p. 107-111.

[13] Koerich, A.L., R. Sabourin, and C.Y. Suen, *Large vocabulary off-line handwriting recognition: A survey.* Pattern Analysis & Applications, 2003. 6(2): p. 97-121.

[14] Al-Emami, S. and M. Usher, *On-line recognition of handwritten Arabic characters.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990. 12(7): p. 704-710.

[15] El-Dabi, S.S., R. Ramsis, and A. Kamel, *Arabic character recognition system: a statistical approach for recognizing cursive typewritten text.* Pattern Recognition, 1990. 23(5): p. 485-495.

[16] Saad, M.K. and W. Ashour. *Osac: Open source arabic corpora*. in *6th ArchEng Int. Symposiums, EEECS*. 2010.

[17] Abed, M.A., *Freeman chain code contour processing for handwritten isolated Arabic characters recognition.* Alyrmook University Magazine,Baghdad, 2012: p. 1-12.

[18] Lorigo, L.M. and V. Govindaraju, *Offline Arabic handwriting recognition: a survey.* IEEE Transactions on Pattern Analysis & Machine Intelligence, 2006(5): p. 712-724.

[19] Zeki, A.M. and M.S. Zakaria. *Challenges in recognizing Arabic characters*. in *The national conference for computer. Abu-al-Aziz king University. Arabia Saudi. April 2004*. 2004.

[20] Abuhaiba, I.S., S.A. Mahmoud, and R.J. Green, *Recognition of handwritten cursive Arabic characters.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994. 16(6): p. 664-672.

[21] Alorifi, F.S., *Automatic identification of Arabic dialects using hidden markov models*. 2008: ProQuest.

[22] Zeki, A.M. *The segmentation problem in arabic character recognition the state of the art*. in *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on*. 2005. IEEE.

[23] Favata, J.T., *Off-line general handwritten word recognition using an approximate beam matching algorithm.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001. 23(9): p. 1009-1021.

[24] Tappert, C.C., C.Y. Suen, and T. Wakahara, *The state of the art in online handwriting recognition.* IEEE Transactions on pattern analysis and machine intelligence, 1990. 12(8): p. 787-808.

[25] O'Gorman, L. and R. Kasturi, *Document image analysis*. Vol. 39. 1995: IEEE Computer Society Press Los Alamitos, CA.

[26] Amin, A., G. Masini, and J. Haton, *Recognition of handwritten Arabic words and sentences.* ICPR84, 1984: p. 1055-1057.

[27] Amin, A. and J.F. Mari, *Machine recognition and correction of printed Arabic text.* IEEE Transactions on systems, man, and cybernetics, 1989. 19(5): p. 1300-1306.

[28] Al-Ma'adeed, S.A., *Recognition of off-line handwritten Arabic words*. 2004: University of Nottingham.

[29] Al-Rashaideh, H., *Preprocessing phase for Arabic word handwritten recognition.* Информационные процессы, 2006. 6(1).

[30] Bieniecki, W., S. Grabowski, and W. Rozenberg. *Image preprocessing for improving ocr accuracy*. in *Perspective*

Technologies and Methods in MEMS Design, 2007. MEMSTECH 2007. International Conference on. 2007. IEEE.

[31] Suliman, A., et al., *Chain Coding and Pre Processing Stages of Handwritten Character Image File.* electronic Journal of Computer Science and Information Technology, 2011. 2(1).

[32] Casey, R.G. and E. Lecolinet, *A survey of methods and strategies in character segmentation.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1996. 18(7): p. 690-706.

[33] Senior, A.W. and A.J. Robinson, *An off-line cursive handwriting recognition system.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998. 20(3): p. 309-321.

[34] Alma'adeed, S., C. Higgens, and D. Elliman. *Recognition of off-line handwritten Arabic words using hidden Markov model approach*. in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. 2002. IEEE.

[35] Kavallieratou, E., N. Fakotakis, and G. Kokkinakis, *Skew angle estimation for printed and handwritten documents using the Wigner–Ville distribution.* Image and Vision Computing, 2002. 20(11): p. 813-824.

[36] Espana-Boquera, S., et al., *Improving offline handwritten text recognition with hybrid HMM/ANN models.* IEEE transactions on pattern analysis and machine intelligence, 2011. 33(4): p. 767-779.

[37] Menasri, F., et al. *Shape-based alphabet for off-line Arabic handwriting recognition*. in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. 2007. IEEE.

[38] Benouareth, A., A. Ennaji, and M. Sellami, *Arabic handwritten word recognition using HMMs with explicit state duration.* Eurasip journal on advances in signal processing, 2007. 2008(1): p. 247354.

[39] Atallah, A.-S. and K. Omar, *Methods of arabic language baseline detection–The state of art.* IJCSNS, 2008. 8(10): p. 137.

[40] Kumar, H. and P. Kaur, *A Comparative Study of Iterative Thinning Algorithms for BMP Images,*2014.

[41] Leibe, B., A. Leonardis, and B. Schiele, *Robust object detection with interleaved categorization and segmentation.* International journal of computer vision, 2008. 77(1-3): p. 259-289.

[42] Musa, M.E., B.A. Bashir, and M.N. Ismail, *Designing an Arabic Handwritten Segmentation System.* International Journal of Computer (IJC), 2016. 20(1): p. 199-209.

[43] Arica, N. and F.T. Yarman-Vural, *An overview of character recognition focused on off-line handwriting.* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2001. 31(2): p. 216-233.

[44] Arica, N. and F.T. Yarman-Vural, *Optical character recognition for cursive handwriting.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. 24(6): p. 801-813.

[45] Rehman, A., D. Mohamad, and G. Sulong, *Implicit vs explicit based script segmentation and recognition: a performance comparison on benchmark database.* Int. J. Open Problems Compt. Math, 2009. 2(3): p. 352-364.

[46] Alabodi, J. and X. Li, *AN EFFECTIVE APPROACH TO OFFLINE ARABIC HANDWRITING RECOGNITION.* International Journal of Artificial Intelligence & Applications, 2013. 4(6): p. 1.

[47] Xiu, P., et al. *Offline handwritten arabic character segmentation with probabilistic model*. in *International Workshop on Document Analysis Systems*. 2006. Springer.

[48] Benouareth, A., A. Ennaji, and M. Sellami. *HMMs with explicit state duration applied to handwritten Arabic word recognition*. in

Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. 2006. IEEE.

[49] Heutte, L., et al., *A structural/statistical feature based vector for handwritten character recognition.* Pattern recognition letters, 1998. 19(7): p. 629-641.

[50] Yousif, I. and A. Shaout, *Off-line Handwriting Arabic Text Recognition: A Survey*.International Journal of Advanced Research in Computer Science and Software Engineering,2014.4(7).

[51] AlKhateeb, J.H., et al., *Multiclass classification of unconstrained handwritten Arabic words using machine learning approaches.* Open Signal Processing Journal, 2009. 2: p. 21-28.

[52] Khorsheed, M.S. and W.F. Clocksin. *Structural Features of Cursive Arabic Script*. in *BMVC*. 1999. Citeseer.

[53] Swiniarski, R.W. and A. Skowron, *Rough set methods in feature selection and recognition.* Pattern recognition letters, 2003. 24(6): p. 833-849.

[54] Swiniarski, R.W. and A. Skowron, *Rough set methods in feature selection and recognition.* Pattern recognition letters, 2003. 24(6): p. 833-849.

[55] Sánchez-Cruz, H., E. Bribiesca, and R.M. Rodríguez-Dagnino, *Efficiency of chain codes to represent binary objects.* Pattern Recognition, 2007. 40(6): p. 1660-1674.

[56] Freeman, H., *Shape description via the use of critical points.* Pattern recognition, 1978. 10(3): p. 159-166.

[57] Suen, C.Y., *Character recognition by computer and applications.* Handbook of pattern recognition and image processing, 1986: p. 569-586.

[58] Leila, C. and B. Mohammed, *ART NETWORK FOR ARABIC HANDWRITTREN RECOGNITION SYSTEM.*

[59] Graves, A. and J. Schmidhuber. *Offline handwriting recognition with multidimensional recurrent neural networks*. in *Advances in neural information processing systems*. 2009.

[60] Bunke, H. and P.S. Wang, *Handbook of character recognition and document image analysis*, 1997, World Scientific.

[61] Märgner, V., H. El Abed, and M. Pechwitz. *Offline handwritten arabic word recognition using hmm-a character based approach without explicit segmentation*. in *Actes du 9ème Colloque International Francophone sur l'Ecrit et le Document*. 2006. SDN06.

[62] Leila, C., K. Maâmar, and C. Salim. *Combining neural networks for Arabic handwriting recognition*. in *Programming and Systems (ISPS), 2011 10th International Symposium on*. 2011. IEEE.

[63] Al-Hajj Mohamad, R., L. Likforman-Sulem, and C. Mokbel, *Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009. 31(7): p. 1165-1177.

[64] Parhami, B. and M. Taraghi, *Automatic recognition of printed Farsi texts.* Pattern Recognition, 1981. 14(1-6): p. 395-403.

[65] Taghva, K., J. Borsack, and A. Condit. *Expert system for automatically correcting OCR output*. in *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*. 1994. International Society for Optics and Photonics.

[66] Plamondon, R. and S.N. Srihari, *Online and off-line handwriting recognition: a comprehensive survey.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000. 22(1): p. 63-84.

[67] Dehghan, M., et al., *Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM.* Pattern Recognition, 2001. 34(5): p. 1057-1065.

[68] Wafa Ali, M.M., *Arabic Handwritten Names Recognition by Using Holistic Approach.* 2009.

[69] Amara, N.E.B. and F. Bouslama, *Classification of Arabic script using multiple sources of information: State of the art and perspectives.* International Journal on Document Analysis and Recognition, 2003. 5(4): p. 195-212.

[70] Sari, T., L. Souici, and M. Sellami. *Off-line handwritten Arabic character segmentation algorithm: ACSA*. in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*. 2002. IEEE.

[71] Pechwitz, M. and V. Maergner. *HMM based approach for handwritten Arabic word recognition using the IFN/ENIT-database*. in *2013 12th International Conference on Document Analysis and Recognition*. 2003. IEEE Computer Society.

[72] Musa, M.E. *Arabic handwritten datasets for pattern recognition and machine learning*. in *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*.

[73] Ali, O.B. and A. Shaout, *Hybrid Arabic Handwritten Character Recognition Using PCA and ANFIS.* International Arab Conference on Information Technology,2016.

[74] Hammad, N.H. and M.E. Musa, *The Impact of Dots Representation in Recognition of Isolated Arabic Characters.* International Journal of Information Engineering & Electronic Business, 2016. 8(6).

[75] Mergani, W.A., *Solving Problems of Thinned Handwritten Arabic words by Zhang and Suen Algorithms*, 2013, Sudan University of Science and Technology.

[76] Ali, O.B., *Use of two Stage Neural Networks for Recognition of Isolated Arabic Optical Characters*, 2011, Sudan University of Science and Technology.

[77] Pechwitz, M., et al. *IFN/ENIT-database of handwritten Arabic words*. in *Proc. of CIFED*. 2002. Citeseer.

[78] Pechwitz, M., et al. *IFN/ENIT-database of handwritten Arabic words*. in *Proc. of CIFED*. 2002. Citeseer.

[79] Mozaffari, S., et al. *A comprehensive isolated Farsi/Arabic character database for handwritten OCR research*. in *Tenth International Workshop on Frontiers in Handwriting Recognition*. 2006. Suvisoft.

[80] Farooq, F., V. Govindaraju, and M. Perrone. *Pre-processing methods for handwritten Arabic documents*. in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. 2005. IEEE.

[81] Dreuw, P., S. Jonas, and H. Ney. *White-space models for offline Arabic handwriting recognition*. in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. 2008. IEEE.

[82] Ball, G.R. and S.N. Srihari. *Semi-supervised learning for handwriting recognition*. in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. 2009. IEEE.

[83] Touj, S.M., N.E.B. Amara, and H. Amiri. *A hybrid approach for off-line Arabic handwriting recognition based on a Planar Hidden Markov modeling*. in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. 2007. IEEE.

[84] Ratzlaff, E.H. *Methods, reports and survey for the comparison of diverse isolated character recognition results on the UNIPEN database*. in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. 2003. IEEE.

[85] Jumari, K. and M.A. Ali, *A survey and comparative evaluation of selected off-line Arabic handwritten character recognition systems.* Jurnal Teknologi, 2012. 36(1): p. 1–18.

[86] El-Glaly, Y. and F. Quek, *Isolated Handwritten Arabic Character Recognition using Multilayer Perceptron and K Nearest Neighbor Classifiers.* 2011.

[87] Rachidi, A., M. El Yassa, and D. Mammass. *A Pretopological approach for handwritten isolated Arabic characters recognition*. in *Proceedings of the second international symposium on communication, control and signal processing, Morocco*. 2006.

[88] Abed, M.A., A.N. Ismail, and Z.M. Hazi, *Pattern recognition using genetic algorithm.* International Journal of Computer and Electrical Engineering, 2010. 2(3): p. 583.

[89] Abandah, G.A., K.S. Younis, and M.Z. Khedher. *Handwritten Arabic character recognition using multiple classifiers based on letter form*. in *Proceeding of 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, Innsbruck, Austria, Feb*. 2008.

[90] Sahlol, A. and C. Suen, *A Novel Method for the Recognition of Isolated Handwritten Arabic Characters.* arXiv preprint arXiv:1402.6650, 2014.

[91] Rabiner, L. and B.-H. Juang, *An introduction to hidden Markov models.* ASSP Magazine, IEEE, 1986. 3(1): p. 4-16.

[92] Rabiner, L.R., *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, 1989. 77(2): p. 257-286.

[93] Plötz, T. and G.A. Fink, *Markov Models for handwriting recognition*. 2012: Springer Science & Business Media.

[94] Baum, L.E., et al., *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains.* The annals of mathematical statistics, 1970: p. 164-171.

[95] Bengio, Y., *Markovian models for sequential data.* Neural computing surveys, 1999. 2(1049): p. 129-162.

[96] Petrushin, V.A. *Hidden markov models: Fundamentals and applications*. in *Online Symposium for Electronics Engineer*. 2000.

[97] Zhou, G. and J. Su. *Named entity recognition using an HMM-based chunk tagger*. in *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 2002. Association for Computational Linguistics.

[98] Forney Jr, G.D., *The viterbi algorithm.* Proceedings of the IEEE, 1973. 61(3): p. 268-278.

[99] Ghahramani, Z. and M.I. Jordan, *Factorial hidden Markov models.* Machine learning, 1997. 29(2-3): p. 245-273.

[100] Stenger, B., et al. *Topology free hidden markov models: Application to background modeling*. in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. 2001. IEEE.

[101] Rabiner, L.R. and B.-H. Juang, *Fundamentals of speech recognition.* 1993.

[102] Ahmed, A.E., *Performance Tests On Several Parametric Representations For An Arabic Phoneme Recognition System Using HMMs.* WIT Transactions on Information and Communication Technologies, 1970. 20.

[103] Maaly, I., A. Elobeid, and K.A. Ahmed, *New parameters for resolving acoustic confusability between Arabic phonemes in a phonetic HMM recognition system.* WIT Transactions on Information and Communication Technologies, 2002. 28.

[104] Boreczky, J.S. and L.D. Wilcox. *A hidden Markov model framework for video segmentation using audio and image features*. in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. 1998. IEEE.

[105] Liu, Z., J. Huang, and Y. Wang. *Classification TV programs based on audio information using hidden Markov model*. in *Multimedia Signal Processing, 1998 IEEE Second Workshop on*. 1998. IEEE.

[106] Alma'adeed, S., C. Higgins, and D. Elliman, *Off-line recognition of handwritten Arabic words using multiple hidden Markov models.* Knowledge-Based Systems, 2004. 17(2): p. 75-79.

[107] Kessentini, Y., T. Paquet, and A.B. Hamadou, *Off-line handwritten word recognition using multi-stream hidden Markov models.* Pattern Recognition Letters, 2010. 31(1): p. 60-70.

[108] Marti, U.-V. and H. Bunke, *Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system.* International journal of Pattern Recognition and Artificial intelligence, 2001. 15(01): p. 65-90.

[109] Khorsheed, M.S., *Recognising handwritten Arabic manuscripts using a single hidden Markov model.* Pattern Recognition Letters, 2003. 24(14): p. 2235-2242.

[110] El-Hajj, R., C. Mokbel, and L. Likforman-Sulem. *Recognition of Arabic handwritten words using contextual character models*. in *Electronic Imaging 2008*. 2008. International Society for Optics and Photonics.

[111] El-Hajj, R., L. Likforman-Sulem, and C. Mokbel. *Arabic handwriting recognition using baseline dependant features and hidden Markov modeling*. in *Document Analysis and Recognition,*

2005. Proceedings. Eighth International Conference on. 2005. IEEE.

[112] Daramola, S.A. and T.S. Ibiyemi, *Offline signature recognition using hidden markov model (HMM).* International journal of computer applications, 2010. 10(2): p. 17-22.

[113] Justino, E.J., et al. *An off-line signature verification system using HMM and graphometric features*. in *Fourth IAPR International Workshop on Document Analysis Systems (DAS), Rio de*. 2000. Citeseer.

[114] Terrapon, N., et al., *Fitting hidden Markov models of protein domains to a target species: application to Plasmodium falciparum.* BMC bioinformatics, 2012. 13(1): p. 1.

[115] Huang, B.Q., Y. Zhang, and M.T. Kechadi. *Preprocessing techniques for online handwriting recognition*. in *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*. 2007. IEEE.

[116] Cheung, A., M. Bennamoun, and N.W. Bergmann, *An Arabic optical character recognition system using recognition-based segmentation.* Pattern recognition, 2001. 34(2): p. 215-233.

[117] Khorsheed, M.S., *Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK).* Pattern Recognition Letters, 2007. 28(12): p. 1563-1571.

[118] Lawgali, A., et al., *Handwritten Arabic character recognition: Which feature extraction method?* International Journal of Advanced Science and Technology, 2011. 34: p. 1-8.

[119] Gatos, B., I. Pratikakis, and S.J. Perantonis, *Adaptive degraded document image binarization.* Pattern recognition, 2005. 39(3).

[120] Otsu, N., *A threshold selection method from gray-level histograms.* Automatica, 1975. 11(285-296): p. 23-27.

[121] Moghaddam, R.F. and M. Cheriet, *AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization.* Pattern Recognition, 2012. 45(6): p. 2419-2431.

[122] Zhang, Y. and L. Wu, *Fast document image binarization based on an improved adaptive Otsu's method and destination word accumulation.* Journal of Computational Information Systems, 2011. 7(6): p. 1886-1892.

[123] Lam, L., S.-W. Lee, and C.Y. Suen, *Thinning methodologies-a comprehensive survey.* IEEE Transactions on pattern analysis and machine intelligence, 1992. 14(9): p. 869-885.

[124] Lam, L. and C.Y. Suen, *An evaluation of parallel thinning algorithms for character recognition.* IEEE Transactions on pattern analysis and machine intelligence, 1995. 17(9): p. 914-919.

[125] Liu, C., Y.-J. Liu, and R. Dai, *Preprocessing and statistical/structural feature extraction for handwritten numeral recognition.* Progress of Handwriting Recognition, World Scientific, Singapore, 1997: p. 161-168.

[126] Melhi, M., S.S. Ipson, and W. Booth, *A novel triangulation procedure for thinning hand-written text.* Pattern Recognition Letters, 2001. 22(10): p. 1059-1071.

[127] Kasturi, R., L. O'gorman, and V. Govindaraju, *Document image analysis: A primer.* Sadhana, 2002. 27(1): p. 3-22.

[128] Märgner, V. and H.E. Abed. *Arabic handwriting recognition competition*. in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. 2007. IEEE.

[129]El-Yacoubi, A., et al., *Off-line handwritten word recognition using hidden markov models.* Knowledge-based intelligent techniques in character recognition, 1999: p. 193-229.

[130]Khan, H.I., *Isolated Kannada Character Recognition using Chain Code Features.* International Journal of Science and Research, 2013.

[131] Yang, M., K. Kpalma, and J. Ronsin, *A survey of shape feature extraction techniques.* Pattern recognition, 2008: p. 43-90.

[132] Sampath, A., C. Tripti, and V. Govindaru, *Freeman code based online handwritten character recognition for Malayalam using backpropagation neural networks.* International journal on Advanced computing, 2012. 3(4): p. 51-58.

[133] Omer, M.A.H. and S.L. Ma. *Online Arabic handwriting character recognition using matching algorithm*. in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*. 2010. IEEE.

[134] Izakian, H., et al., *Multi-font Farsi/Arabic isolated character recognition using chain codes.* World Academy of Science, Engineering and Technology, 2008. 43: p. 67-70.

[135] Al-Badr, B. and S.A. Mahmoud, *Survey and bibliography of Arabic optical text recognition.* Signal processing, 1995. 41(1): p. 49-77.

[136] Al-Muhtaseb, H.A., S.A. Mahmoud, and R.S. Qahwaji, *Recognition of off-line printed Arabic text using Hidden Markov Models.* Signal Processing, 2008. 88(12): p. 2902-2912.

[137] Bunke, H., S. Bengio, and A. Vinciarelli, *Offline recognition of unconstrained handwritten texts using HMMs and statistical language models.* IEEE transactions on Pattern analysis and Machine intelligence, 2004. 26(6): p. 709-720.

[138] Günter, S. and H. Bunke, *HMM-based handwritten word recognition: on the optimization of the number of states, training*

iterations and Gaussian components. Pattern Recognition, 2004. 37(10): p. 2069-2079.

[139] KESSENTINI, Y., T. PAQUET, and A. BENHAMADOU, *A multi-stream HMM-based approach for off-line multi-script handwritten word recognition.* a a, 2008. 1: p. 1.

[140] Gellert, A. and L. Vintan, *Person movement prediction using hidden Markov models.* Studies in Informatics and control, 2006. 15(1): p. 17.

[141] Lorigo, L. and V. Govindaraju. *Segmentation and pre-recognition of Arabic handwriting*. in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. 2005. IEEE.

[142] Pratap, R., *Getting started with MATLAB.* A quick introduction for scientists and engineers. Oxford University, 1996.

[143] Solomon, C. and T. Breckon, *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. 2011: John Wiley & Sons.

[144] Murphy, K., *Hidden markov model (hmm) toolbox for matlab.* online at http://www. ai. mit. edu/~ murphyk/Software/HMM/hmm. html, 2005.

[145] Sahloul, A. and C. Suen. *OFF-line system for the recognition of handwritten arabic character*. in *Fourth International Conference on Computer Science & Information Technology*. 2014.

[146] McAndrew, A., *An introduction to digital image processing with matlab notes for scm2511 image processing.* School of Computer Science and Mathematics, Victoria University of Technology, 2004: p. 1-264.

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