Arabic Text Recognition of Printed Manuscripts

Efficient Recognition of Off-Line Printed Arabic Text Using Hidden Markov Models,

Bigram Statistical Language Model, and Post-Processing

By

Husni Abdulghani Al-Muhtaseb

BSc, MSc

A Thesis Submitted for the Degree of Doctor of Philosophy

2010

Digital Imaging

School of Computing, Informatics and Media

University of Bradford

Abstract

Arabic text recognition was not researched as thoroughly as other natural languages. The need for automatic Arabic text recognition is clear. In addition to the traditional applications like postal address reading, check verification in banks, and office automation, there is a large interest in searching scanned documents that are available on the internet and for searching handwritten manuscripts. Other possible applications are building digital libraries, recognizing text on digitized maps, recognizing vehicle license plates, using it as first phase in text readers for visually impaired people and understanding filled forms.

This research work aims to contribute to the current research in the field of optical character recognition (OCR) of printed Arabic text by developing novel techniques and schemes to advance the performance of the state of the art Arabic OCR systems.

Statistical and analytical analysis for Arabic Text was carried out to estimate the probabilities of occurrences of Arabic character for use with Hidden Markov models (HMM) and other techniques.

Since there is no publicly available dataset for printed Arabic text for recognition purposes it was decided to create one. In addition, a minimal Arabic script is proposed. The proposed script contains all basic shapes of Arabic letters. The script provides efficient representation for Arabic text in terms of effort and time.

Based on the success of using HMM for speech and text recognition, the use of HMM for the automatic recognition of Arabic text was investigated. The HMM technique adapts to noise and font variations and does not require word or character segmentation of Arabic line images.

In the feature extraction phase, experiments were conducted with a number of different features to investigate their suitability for HMM. Finally, a novel set of features, which resulted in high recognition rates for different fonts, was selected.

The developed techniques do not need word or character segmentation before the classification phase as segmentation is a byproduct of recognition. This seems to be the most advantageous feature of using HMM for Arabic text as segmentation tends to produce errors which are usually propagated to the classification phase.

Eight different Arabic fonts were used in the classification phase. The recognition rates were in the range from 98% to 99.9% depending on the used fonts. As far as we know, these are new results in their context. Moreover, the proposed technique could be used for other languages. A proof-of-concept experiment was conducted on English characters with a recognition rate of 98.9% using the same HMM setup. The same techniques where conducted on Bangla characters with a recognition rate above 95%.

Moreover, the recognition of printed Arabic text with multi-fonts was also conducted using the same technique. Fonts were categorized into different groups. New high recognition results were achieved.

To enhance the recognition rate further, a post-processing module was developed to correct the OCR output through character level post-processing and word level post-processing. The use of this module increased the accuracy of the recognition rate by more than 1%.

Keywords: Arabic text recognition, Hidden Markov Models, Feature extraction, Omni font recognition, Minimal Arabic script.

Table of Contents

Acknowledgements

I would like to express my sincere and deep thanks to my Supervisor Dr. Rami Qahwaji for support, guidance, help and encouragement throughout this research work. I wonder if Dr. Qahwaji from now onward will supervise a research work for a person reaching 50 years. I am also grateful to Professor Sabri Mahmoud for his guidance and support throughout the first two years of my study. Professor Mahmoud also has the credit of suggesting the research topic, suggesting the University and connecting me with my supervisor.

Thanks to the University of Bradford and to the School of Computing, Informatics and Media for providing research resources throughout my stay in Bradford.

This work has greatly benefited from the support of King Fahd University of Petroleum and Minerals (KFUPM). Without KFUPM support this work would not be easily completed. Research resources at KFUPM were also available during the whole period of my study. Thank you KFUPM and sincere thanks to KFUPM management.

Special thanks to Sameh Awaida for his suggestions, advice, and technical help. He saved me huge amount of time through our frequent communications. I would like also to thank Dr. Mohammad Khorsheed of King Abdulaziz City for Science and Technology (KACST) for providing initial data and resources that stimulated this research work.

I would like to thank my sisters and brothers for continuous encouragement and moral support. Their caring calls and messages used to always have beneficial effects. Special thanks to my brother Hamza for his efforts in supporting my family while I am abroad.

Special thanks to AbdolRauf Alhaddad who was my local consultant during my stay at Bradford. Other than his valuable friendship and moral support, his knowledge of the town and the country helped me a lot.

Dedication

To my family,
my beloved wife: Fathiya,
my lovely daughters Lama, Sara and Al-Shayma,
my lovely sons Asem, Mohammad and Amer.

List of Figures

List of Tables

List of Acronyms

ANN	Artificial Neural Network
ASMO	Arab Standardization and Metrology Organization
BBN BYBLOS	A Speech Recognition System
CD-ROM	Compact Disc Read-Only Memory
CPU	Central Processing Unit
DARPA	Defense Advanced Research Projects Agency
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
HTML	Hyper Text Markup Language
ICA	Independent Component Analysis
ICDAR	International Conference on Document Analysis and Recognition
ICFHR	International Conference on Frontiers in Handwriting Recognition
ICPR	International Conference on Pattern Recognition
ISO	International Standards Organization
IWFHR	International Workshop on Frontiers in Handwriting Recognition
KACST	King Abdulaziz City for Science and Technology
KFUPM	King Fahd University of Petroleum and Minerals
LDA	Linear Discriminant Analysis
ML	Maximum Likelihood
MLP	Multi-Layer Perceptron
NN	Neural Network
OCR	Optical Character Recognition
ORAN	Offline Recognition of Arabic Numerals
PATS	Printed Arabic Text Set
PC	Personal Computer
PCA	Principal Component Analysis
PDF	Probability Density Function
RECAM	An Arabic handwritten Recognition System
SVM	Support Vector Machine
VQ	Vector Quantization
WER	Word Error Rate
WMR	Word Model Recognizer

List of Publications

- Husni A. Al-Muhtaseb and Rami S. Qahwaji, “New Advances in the Optical Character Recognition of Arabic text”, In “*Applied Signal and Image Processing: Multidisciplinary Advancements*”, Editors: Dr Rami Qahwaji, Roger Green, Evor Hines, IGI 2010, (Accepted).

- Husni A. Al-Muhtaseb and Rami S. Qahwaji, “A Single Feature Extraction Algorithm for Text Recognition of Different Families of Languages”, *Third Mosharaka International Conference on Communications, Computers and Applications MIC-CCA2009*, Amman, Jordan, October 2009.
- Husni A. Al-Muhtaseb, S. Mahmoud, and Rami S. Qahwaji, “A Novel Minimal Arabic Script for Preparing Databases and Benchmarks for Arabic Text Recognition Research”, *8th WSEAS International Conference on Signal Processing (SIP '09)*, Istanbul, Turkey, June 2009
- Husni A. Al-Muhtaseb, S. Mahmoud, Rami S. Qahwaji, "A Novel Minimal Script for Arabic Text Recognition Databases and Benchmarks", *International Journal of Circuits, Systems and Signal Processing*, Issue 3, Volume 3, 2009, pp. 145-153.
- Husni A. Al-Muhtaseb, S. Mahmoud, and Rami S. Qahwaji, “Automatic Arabic Text Image Optical Character Recognition Method”, Patent Pending, Filed in USA 12/382916, April 2009.
- Husni A. Al-Muhtaseb, S. Mahmoud, and Rami S. Qahwaji, “Recognition of Off-line printed Arabic text Using Hidden Markov Models”, *Signal Processing*, Volume 88, Issue 12, December 2008, pp. 2902-2912.
- Husni Al-Muhtaseb, S. Mahmoud, R. Qahwaji, “A Statistical Analysis to Support Arabic Text Recognition”, *(in Arabic) the International Symposium on Computers and Arabic Language*, Riyadh, November 2007.

Chapter 1. Introduction

One way to avoid retyping a scanned document is to use an optical character recognition tool to convert the text images in the scanned document into an editable text. Such a tool takes the scanned document as a picture and recognizes the text in the picture and makes it available in text format.

Optical Arabic cursive text recognition has received renewed research interest following recent successes in optical character recognition for other languages. Arabic text recognition, which was not researched as thoroughly as Latin, Chinese, or Japanese, is receiving more attention from both Arabic and non-Arabic-speaking researchers.

Irrespective of the language under consideration, some traditional applications of text recognition include: check verification, office automation, reading postal address, writer identification, and signature verification. Searching scanned documents available on the internet and searching Arabic historical manuscripts are also emerging applications. When Arabic is considered, the need to advance each one of these applications is serious as there is a lack of real applications in these areas.

Arabic is the first language for more than 400 million people in the world [*1*]. It is also used by more than triple the previous number of Muslims all over the world as a second language, for it is the language in which the Holy Qur'an was revealed. That is, Arabic is being used by more than 1.5 billion people. Arabic was added to the official languages of the United Nations in 1973 as the sixth language. The other five official languages (Chinese, English, French, Russian and Spanish) were chosen when the United Nations was founded [*2*] [*3*]. Also as has been reported by National Geographic [*4*], Arabic is expected to be one of the 5 major languages by 2050.

Arabic is one of the Semitic languages. The Arabic script is being used/has been used in other languages. Some of which are Hausa, Kashmiri, Kazak, Kurdish, Kyrghyz, Malay, Morisco, Pashto, Persian/Farsi, Punjabi, Sindhi, Tatar, Turkish, Uyghur, and Urdu [*5*].

This chapter is organized as follows. Section 1.1 describes briefly the general phases of an Arabic optical character recognition systems (OCR). Section 1.2 presents some characteristics of Arabic Text. Section 1.21.3 introduces the motivation behind this research work. The domain of the addressed problem is presented in Section 1.4. The objectives of the research are summarized in Section 1.5. Section 1.6 presents the structure of the thesis.

1.1 Automatic Arabic Text Recognition

A generic model for an automatic Arabic text recognition system is shown in Figure 1.1. The automated process starts by scanning an image of an Arabic text. The scanned image is analyzed in the pre-processing phase to improve its condition. The pre-processing phase might include noise removal, skew/slant detection and correction and normalization.

Usually, the text image is segmented into images of lines. Depending on the used feature extraction and classification techniques, a character-based segmentation phase may or may not be necessary. Since Arabic text is cursive, some techniques require the segmentation of Arabic text before the feature extraction phase. During segmentation, the Arabic text image is segmented into lines. Furthermore, the line images could be segmented into words/sub-words and then to characters or even sub-characters based on the used technique. If the image under consideration contains tables and figures, then their text is extracted for recognition.

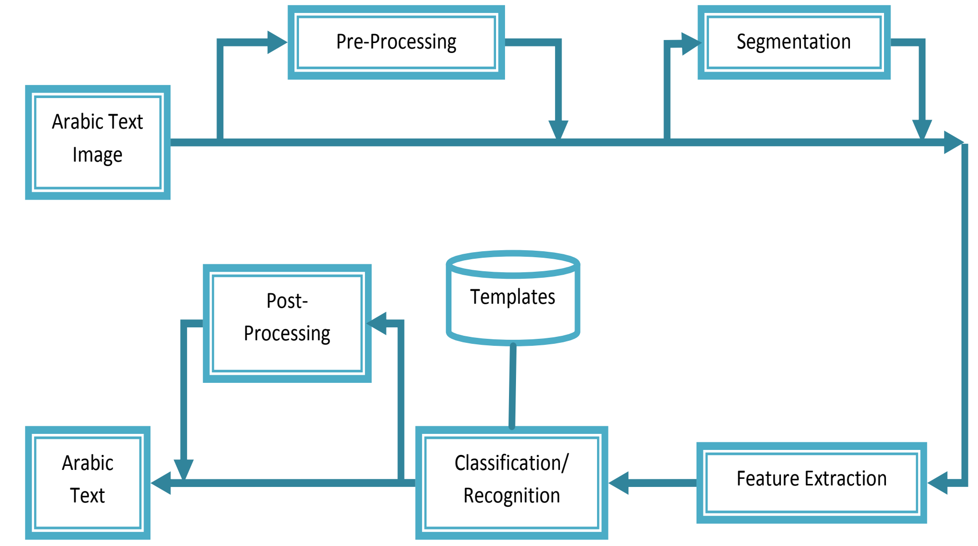


Figure 1.1: Optical text recognition architecture.

The feature extraction phase is applied to a line, a word, a sub-word, a character, or sub-character based on the method used. The features are extracted from basic units (a word, a sub-word, a character, or sub-character) and used in classification and recognition. The actual recognition is done through the classification/recognition phase that produces text representation of sequences of words, sub-words, or characters that represent the text image. The representations of these basic units could be saved in different formats (plain Unicode text, HTML, PDF ...). The post-processing phase is usually based on a spell-checking tool that possibly adds more accuracy to the resulting recognized text.

1.2 Characteristics of Arabic Text

Arabic is a cursive language written from right to left. It has 28 basic letters. An Arabic letter might have up to four different shapes depending on the position of the letter in the word: whether it is a standalone letter, connected only from right (initial form), connected only from left (terminal form), or connected from both sides (medial form). Letters of a word may overlap vertically (even without touching).

Arabic letters do not have fixed size (height and width). Letters in a word can have diacritics (short vowels) such as *Fat-hah, Dhammah, Shaddah, sukoon* and *Kasrah*. Moreover, *Tanween* may be formed by having double *Fat-hah*, double *Dhammah*, or double *Kasrah.* Figure 1.2 lists these diacritics. These diacritics are written as strokes, placed either on top of, or below, the letters. A different diacritic on a letter may change the meaning of a word. Readers of Arabic are used to reading un-vocalized text by deducing the meaning from context.

Fat-hah ـَ	Dhammah ـُ	Shaddah ـّ
Kasrah ـِ	Sukoon ـْ	TanweenFat-h ـً
Tanween Dhamm ـٌ		Tanween Kasr ـٍ

Figure 1.2: Arabic short vowels (diacritics)

Figure 1.3 shows some of the characteristics of Arabic text. It shows a base line, overlapping letters, diacritics, and two shapes of *Noon* character (initial and medial).

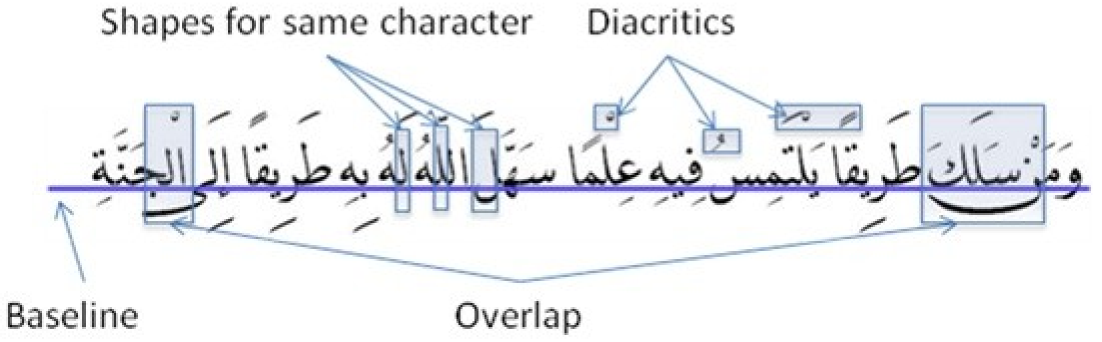

Shapes for same character
Diacritics
ومن سلك طريقا يلتمس فيه علما سهل الله له به طريقا إلى الجنة
Baseline
Overlap

Figure 1.3: An example of an Arabic sentence indicating some characteristics of Arabic text.

As Arabic numbers are not connected and are used globally, we concentrated our work on Arabic letters throughout this thesis. As we stated earlier, Arabic has 28 main letters as shown in Figure 1.4. When considering presenting Arabic characters to computers, some of the main letters have been extended into separate letters for ease of presentations and usability by the Arab Standardization and Metrology Organization (ASMO). The standard Arabic codepages (character sets) ASMO-449, ASMO-708 and ISO 8859-6 define 36 Arabic letters (see Figure 1.5). When OCR is considered, it is needed to add *Lam-Alef* in its 4 different forms. Although *Lam-Alef* is a sequence of two alphabets, they are written as one set. This sequence should be treated as one set. So, four more sets should be added to the alphabets; one with bare *Alef*, the second with *Alef-Maddah*, the third with *Alef-up-Hamza* and the fourth with *Alef-down-Hamza* as shown in Figure 1.6. This expands the number of Arabic letters to 40. Each alphabet can take different numbers of shapes (from 1 to 4). Hence, the total number of shapes is 125 (one letter has only one shape, others have two, and the most have four shapes).

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي

Figure 1.4: Basic Arabic 28 letters.

ء آ أ ؤ إ ئ ا ب ة ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ى ي

Figure 1.5: Extended Arabic letters by ASMO.

ء آ أ ؤ إ ئ ا ب ة ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و لآ لأ لإ
لا ى ي

Figure 1.6: Expanded Arabic alphabets by adding different versions of Lam-Alef sequences.

Table 1-1 shows the basic Arabic letters with their categories. They are grouped into 3 different classes according to the number of shapes a letter takes. The first Class (class 1) consists of a single shape of the *Hamza* which comes in stand-alone state

(Number 1 in Table 1-1). *Hamza* does not connect with any other letter. The second class (class 2) presents the letters that can come either standalone or connected only from right (medial category). This class consists of *Alef Madda, Alef up Hamza, Waw Hamza, Alef down Hamza, Alef, Tah Marboutah, Dal, Dhal, Ra, Zain, Waw, Lam Alef Madda, Lam Alef Hamza up, Lam Alef Hamza down,* and *Lam Alef* (numbers 2-5, 7, 9, 15-18, and 35-39 in Table 1-1). The third class (class 3) consists of the letters that can be connected from either side or both sides as well as they can appear as standalone. This class consists of *Hamza Kursi, Baa, Taa, Thaa, Jeem Haa, Khaa, Seen, Sheen, Sad, Dhad, Dhaa, THaa, Ain, Gain, Faa, Qaaf, Kaaf, Laam ,Meem, Noun, Haa, Yaa* (numbers 6, 8, 10-14, 19-33, and 40 in Table 1-1). Table 1-2 shows a summary of these classes.

Although an Arabic letter might have up to 4 different shapes, each letter is saved using only one code. It is the duty of a built-in driver to make contextual analysis to decide the right shape to display, depending on the previous and next characters if available. When it is needed to consider different shapes of Arabic letters for a given Arabic text file, a contextual analysis algorithm is needed. Such algorithm takes the letter, its predecessor, and its successor and identifies the right shape depending on the classes of the letters.

Table 1-2: Basic shapes of Arabic letters.

no	Stand-alone	Term.	Medial	Initial	Shapes	Class
1	ء	ء	ء	ء	1	1
2	آ	ـآ	ـآ	آ	2	2
3	أ	ـأ	ـأ	أ	2	2
4	ؤ	ـؤ	ـؤ	ؤ	2	2
5	إ	ـإ	ـإ	إ	2	2
6	ئ	ـئ	ـئـ	ئـ	4	3
7	ا	ـا	ـا	ا	2	2
8	ب	ـب	ـبـ	بـ	4	3
9	ة	ـة	ـة	ة	2	2
10	ت	ـت	ـتـ	تـ	4	3
11	ث	ـث	ـثـ	ثـ	4	3
12	ج	ـج	ـجـ	جـ	4	3
13	ح	ـح	ـحـ	حـ	4	3
14	خ	ـخ	ـخـ	خـ	4	3
15	د	ـد	ـد	د	2	2
16	ذ	ـذ	ـذ	ذ	2	2
17	ر	ـر	ـر	ر	2	2
18	ز	ـز	ـز	ز	2	2
19	س	ـس	ـسـ	سـ	4	3
20	ش	ـش	ـشـ	شـ	4	3
21	ص	ـص	ـصـ	صـ	4	3
22	ض	ـض	ـضـ	ضـ	4	3
23	ط	ـط	ـطـ	طـ	4	3
24	ظ	ـظ	ـظـ	ظـ	4	3
25	ع	ـع	ـعـ	عـ	4	3
26	غ	ـغ	ـغـ	غـ	4	3
27	ف	ـف	ـفـ	فـ	4	3
28	ق	ـق	ـقـ	قـ	4	3
29	ك	ـك	ـكـ	كـ	4	3
30	ل	ـل	ـلـ	لـ	4	3
31	م	ـم	ـمـ	مـ	4	3
32	ن	ـن	ـنـ	نـ	4	3
33	ه	ـه	ـهـ	هـ	4	3
34	و	ـو	ـو	و	2	2
35	لآ	لآ	ـلآ	لآ	2	2
36	لأ	لأ	ـلأ	لأ	2	2
37	لإ	لإ	ـلإ	لإ	2	2
38	لا	لا	ـلا	لا	2	2
39	ى	ـى	ـى	ى	2	2
40	ي	ـي	ـيـ	يـ	4	3

Table 1-1: Classes of Arabic letters depending on number of possible basic shapes.

Class	# of possible shapes	Letters
1	1	ء
2	2	آ أ ؤ إ ا ة د ذ ر ز و لآ لأ لإ لا ى
3	4	ئ ب ت ث ج ح خ س ش ص ض ط ظ ع غ ف ق ك ل م ن ه ي

1.3 Motivation

The advances in text recognition for other languages encouraged the author to investigate techniques for use with Arabic text recognition.

Arabic text is cursive and hence most published work on Arabic text assumes that the text is segmented or applies a segmentation phase to Arabic text before recognition. Segmentation of cursive text, including Arabic, is error prone as has been demonstrated in published work and can be concluded from the characteristics of cursive text (See Bunke and Varga [*6*], Al-Ohali et al. [*7*], and Hu et al. [*8*]). In addition, the errors in the segmentation phase results in more errors in the classification phase.

The special characteristics of Arabic text and the lack of available data and basic tools [*9*] [*10*] increased the motivation to conduct this research work. Moreover, the uncertain road for possible successful outcomes for automatic Arabic text recognition made it challenging. In addition, a successful Arabic OCR may facilitate the way for many applications such as: document automation, writer identification and mobile applications.

1.4 Problem Domain

In this research work the problem of automatic recognition of printed Arabic text is addressed. The emphasis in this work is on the feature extraction and classification phases as these phases have more research potential with respect to automatic Arabic text recognition. Moreover, feature extraction schemes along with the classification phase have crucial effects on the recognition accuracy of OCR systems.

Since Arabic text is cursive and the segmentation of Arabic is an error-prone task, segmentation is widely considered to be the bottleneck in these approaches as errors in segmentation will lead to errors in the classification stage (See Rashwan et al. [*11*], Vinciarelli et al. [*12*]). If a Hidden Markov Models (HMM) technique is used, there would be no need to segment Arabic text to words, sub-words, or characters as segmentation is a by-product of HMM classification. The features of Arabic text line image are extracted and supplied to the HMM in the training and classification tasks. The segmentation is a by-product of the classification. Of course the need to segment the document image into images of lines is still there. However, it is less error-prone. The success of HMM in speech and English character recognition, including handwritten text, make it a good prospect to investigate the technique for Arabic text recognition.

1.5 Objectives

The objective is to address long standing problems in automatic printed Arabic text recognition and develop techniques and procedures to efficiently recognize printed Arabic text. We are mainly addressing the feature extraction and classification phases.

To achieve this objective, the following sub-objectives are addressed:

- Statistical and syntactical analysis for Arabic text will be pursued. The resulting analysis will allow better understanding of suitable feature extraction techniques. The analysis could also be utilized in classifications and post-processing.
- The development of the first public benchmark data for printed Arabic text recognition, as there is no freely available database benchmark for printed Arabic text recognition.

- Developing an efficient extraction technique that leads to more accurate classifications to be used for Arabic text recognition. The target technique aims to be simple and represent the images while keeping the uniqueness of different characters in the image to help in accurate classifications.
- Proposing an efficient recognition technique that is segmentation free to be used along with the developed feature extraction technique.
- Developing post-processing techniques that could enhance the results of an Arabic OCR system.

1.6 Structure of the Thesis

The remaining parts of this thesis are structured as follows.

Chapter 2 *Literature Review:* The main purpose of the literature review is to provide definitions, context, and a clearer understanding of previous research in printed Arabic text recognition. The review highlights some examples of how different types of techniques are being used in the addressed field. It reviews the state-of-the-art, recent advances and limitations in the Arabic text recognition.

Chapter 3 *Statistical Analysis and Data Preparation:* This chapter reports the Statistical analysis for Arabic Text that is carried out to estimate the probabilities of occurrences of Arabic characters for possible use with HMM and other techniques. The chapter also addresses Arabic data preparation. Since there are no adequate dataset benchmarks for printed Arabic text recognition research, work towards making our own data for the research is addressed. Related issues in preparing such database are addressed. In this chapter, a novel minimal set of Arabic characters that could provide efficient representation for Arabic text is presented. This minimal set facilitates the generation of data for use in automatic Arabic text recognition and has reduced the effort and time required.

Chapter 4 *Feature Extraction:* This chapter introduces the new proposed family of schemes for extracting features suitable to be used in HMM-based training and classifications techniques. Different versions of the proposed technique are described. Although the schemes were developed for Arabic text, experiments showed that they could be used for other languages as they preserve the general structure of the images under consideration.

Chapter 5 *Training and Classification for Single Fonts:* The training and Classification phase is presented in this chapter. In addition, results for single font classification are presented. Eight fonts are used and for each font classification results and analysis are presented.

Chapter 6 *Multi-font Classifications and Work with other Languages:* This chapter presents multi-font training and classification results. It also presents the classification of English and Bangla text using the same proposed techniques. The datasets used with each language are presented and the results are shown with analytical discussion.

Chapter 7 *Post-Processing:* This chapter presents the post-processing techniques that have been used to enhance the results of the recognition processes. It introduces a new flexible prototype for OCR post-processing based on character level post-processing and word level post-processing using the knowledge learned from the analysis of Arabic text recognition classifications.

Chapter 8 *Conclusion and Future Work:* The contributions of this research work to the field of Arabic text recognition are presented in this chapter. Possible future research directions in related areas are also discussed.

Chapter 2. Literature Review

2.1 Introduction

Arabic text recognition systems can be divided into two categories: Handwritten text recognition and printed text recognition. The handwritten recognition systems can be categorized into online recognition and offline recognition. On-line recognition aims to recognize the characters while the writer is writing on a tablet using a stylus (See Mezghani et al. [*13*], Manfredi et al. [*14*], Halavati et al. [*15*]). Arabic recognition systems can also address special purpose data such as numerals only, isolated character only, postal address, or literals numbers. The systems can also address cursive open vocabulary text such as cursive letters and letters, numerals and punctuations. Figure 2.1 shows these types of addressed data.

This chapter discusses the state-of-the-art in Arabic text recognition technology. Section 2.2 starts the literature review by surveying related works and printed Arabic OCR techniques. Available databases for Arabic OCR research are discussed in Section 2.3. Related research on pre-processing text images is discussed in Section 2.4. Section 2.5 addresses the literature on segmentation of Arabic Text. Common feature extraction techniques are presented in Section 2.6. Section 2.7 discusses the use of HMM in Arabic text recognition. The state-of-the-art in post-processing is reviewed in Section 2.8. Section 2.9 lists available Commercial Arabic OCR Software. The last section of this chapter (Section 2.10) is a summary and an introduction to the research work behind this thesis.

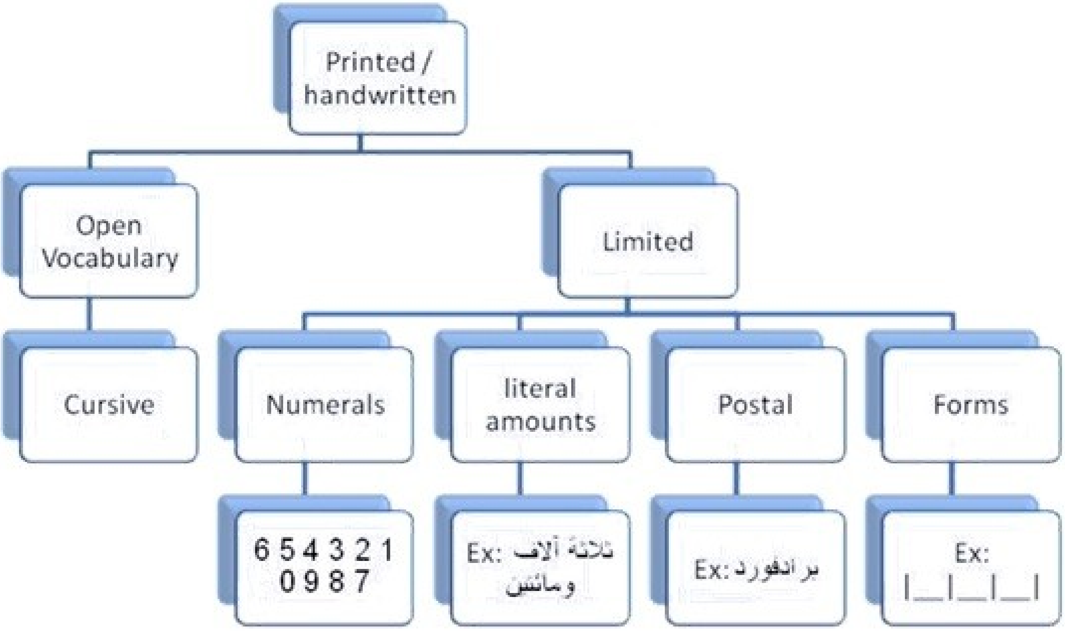


Figure 2.1: OCR addressed data types.

2.2 Surveys and Systems

Early reviews covering Arabic text recognition can be found in [*16*] [*17*]. More recent reviews can be also found in Lorigo and Govindaraju [*18*], Nabawi and Mahmoud [*19*], Haraty and Ghaddar [*20*], Trenkle et al. [*21*], Abandah and Khedher [*22*], Darwish K. [*23*], Ball [*24*], Klassen [*25*], Al-Sulaiti [*10*], Burrow [*26*], AL-Shatnawi, and Omar [*27*], Aburas and Gumah [*28*] and Nikkhou and Choukri [*29*].

Other publications have reported prototype systems for Arabic text/character recognition. The ORAN system reported by Zidouri et al. [*30*] [*31*] was based on Nask font and a recognition rate of 97.5% was reported.

RECAM reported by Sari and Sellami [*32*] is a cursive Arabic handwritten script recognition system using word segmentation. An Arabic printed text recognition using neural networks was suggested by Sarfraz et al. in [*33*]. A multi-font recognition system of printed Arabic text using the BYBLOS speech recognition system was reported by LaPre et al. in [*34*]. Hamami and Berkani [*35*] introduced a multi-font multi-size recognition system for printed Arabic characters. The system is based on the detection of holes and concavities. Gillies et al. [*21*] [*36*] presented a printed Arabic text recognition system with recognition rate of 93% for high quality documents and 89% for lower quality documents.

A recognition system for isolated Arabic characters was reported by Cowell and Husain in [*37*]. Cheung et al. [*38*] presented an Arabic single-font recognition system with 85% accuracy. A system with 90% accuracy was reported by Cheung et al. in [*39*]. An online Arabic handwritten recognition system was presented by the same group of researchers in [*40*]. Aburas and Rehiel [*41*] introduced a Wavelet Compression based system for Off-line Omni-style Handwriting Arabic Character Recognition with a recognition rate of 97% in some cases.

An Arabic OCR system that uses a histogram clustering method for the segmentation of Arabic words has been reported with recognition accuracy of 91.5% by Syiam et al. [*42*]. Feature extraction in the reported system was based on a combination of principle component analysis (PCA) and geometric features of characters. The classifier was designed using a decision tree induction algorithm and Multi-layered Perceptron network (MLP). 65% accuracy was reported by Dehghan et al. in a system that recognized Farsi handwritten words using discrete HMM in [*43*].

Bentouns and Batouche [*44*] proposed the use of support vector machines (SVM) for handwritten Arabic character recognition. Topological and statistical features were extracted to construct vectors. A multi-font Arabic OCR system using Hough transform for feature extraction and Hidden Markov Models for classifications with 96.8% recognition rate, in some cases, was reported by Ben Amor and Ben Amra in [*45*]. Bazzi et al. [*46*] reported an earlier system that could be used for recognition of English and Arabic printed text. They reported an accuracy rate of 95% for specific DARPA data.

2.3 Databases

A few Arabic databases with limited content are available for research in Arabic text recognition. Some of them have been prepared for specific domains and applications such as cheques, numerals contents, and postal addresses. Farah et al. have used Arabic literal amounts (words representing numbers) of 4800 words [*47*]. A database consisting of 26,459 Arabic names, presenting 937 Tunisian town/village names, handwritten by 411 different writers was presented by Pechwitz and Maergner in [*48*] [*49*] and used in several research experiments including Pechwitz et al. [*50*] and Margner et al. [*51*]. A database prepared from text involving 100 persons, where each person wrote 67 literal numbers, 29 of the most popular words in Arabic, three sentences representing numbers and quantities used in cheques, and a free subject chosen by the writer (around 4700 handwritten words) was reported by Al-Ma'adeed et al. in [*52*] [*53*] [*54*]. Alotaibi presented a small database for digits. This database involved 17 persons who each wrote 10 digits 10 times [*55*]. An Arabic and Persian database of isolated characters consisting of 220,000 handwritten forms filled in by more than 50,000 writers was presented by Soleymani and Razzazi in [*56*]. The databases by Al-Ohali et al. in [*7*] and [*57*] contained 29,498 images of sub-words, 15,175 images of Indian-Arabic digits and image samples of both legal and courtesy amounts taken from 3000 real-life bank cheques. Another database for bank cheques included 70 words of Arabic literal amounts extracted from 5000 cheques written by 100 persons was introduced by Maddouri et al. in [*58*]. An automatically generated printed database of 946 Tunisian town names is discussed by Margner and Pechwitz in

[*59*]. Hamid and Haraty used 360 handwritten addresses of around 4000 words [*60*]. The addresses were collected from students and staff at the Lebanese American University, Lebanon. Dehghan et al. [*43*] Presented a database consisting of more than 17820 names of 198 cities in Iran. Kharma et al. presented a general database with signatures which has 37,000 words, 10,000 digits, 2,500 signatures, and 500 free-form Arabic sentences [*61*]. A small isolated character database consisting of 50 images for each character written by 5 persons was introduced by Wanas et al. in [*62*]. Each person wrote the whole 28-character alphabet ten times. DARPA Arabic Corpus consists of 345 scanned pages of printed text in 4 different fonts [*63*]. The system of Bazzi et al. [*46*] used 40 pages of the DARPA database to test their suggested recognition methodology. The research presented by Trenkle et al. in [*64*] used 700 digitized pages from 45 printed documents. The segmentation work by Melhi in [*65*] was based on around 240 digitized pages written by 178 persons. Each person wrote one or two pages of 10 previously prepared text of 13 lines per page.

A technique to automatically generate a database for OCR systems was presented in [*59*]. The technique which was designed to generate an English database for OCR systems was modified and used to generate Arabic Tunisian town names. Generating printed text databases automatically assures 100% correctness of the ground truth information and allows the construction of large databases. A database for the OCR of Arabic printed and handwriting text was introduced by Ben Amara et al. in [*66*]. The database includes images of text phrases, words/sub-words, isolated characters, digits, and signatures. A Second Database for Handwritten Arabic Words, Numbers, and Signatures for OCR was described by Kharma et al. in [*61*].

2.4 Pre-processing

Different pre-processing classes have been proposed for different tasks including normalization, slope correction, slant correction and thinning, see for example Al-Ma'adeed et al. work [*53*]. Sari et al., in [*67*] and [*32*], used a statistical based smoothing algorithm for smoothing and noise reduction. Sarfraz et al. [*33*] [*68*] introduced pre-processing techniques for the removal of isolated pixels, skew detection and correction.

A baseline estimation of handwritten words was described by Pechwitz and Margner in [*69*] where features related to the baseline were examined. Khorsheed and Clocksin [*70*] used Stentiford's algorithm for thinning. Al-Khatib and Mahamud [*71*] addressed removing curvature effects, tilt/skew correction, and noise filtering. Another scheme for tilt correction was introduced by Sarfraz and Shahab in [*72*]. This technique was based on finding the character *Alef* in the image and detecting the skew angle.

A transform technique (Hough Transform) known for its ability to handle distortions and noise was used by Touj et al. for recognition of Arabic printed characters in [*73*] [*74*] [*75*]. Mahmoud [*76*] used normalized Fourier descriptors for Arabic OCR along with contour analysis. The contour of the primary part of the character, the dot, and the hole were extracted. Then Fourier descriptors were computed and used for training. The normalized Fourier descriptors technique is invariant to scale, rotation, and translation. However, there is a trade-off between the gained accuracy and the processing speed. Zahour et al. introduced another contour based method to extract text-lines [*77*]. This method was based on a partial contour tracing algorithm. It was known to be slant sensitive.

A thinning algorithm based on clustering the data image using neural network was used by Altuwaijri and Bayoumi in [*78*]. M. Shirali-Shahreza and S. Shirali-Shahreza have concluded that when removing noise from Arabic text images, care should be taken not to remove dots that are part of the Arabic script [*79*]. A thinning algorithm for poor quality Arabic text images was introduced by Cowell and Hussain in [*80*].

2.5 Segmentation

Zidouri et al. presented a printed Arabic character segmentation based on adaptive dissection. They reported that the system showed promising results with some problems related to character overlapping and ligatures [*81*]. Zheng et al. performed line segmentation as well as word and sub-word segmentation [*82*] using horizontal histograms. However, character segmentation was based on the analysis of the upper contour of the sub-word under consideration. Similar techniques were used by Sari and Sellami [*32*], Romeo-Pakker et al. [*83*], and Olivier et al. [*84*].

Several research techniques bypass the error-prone segmentation phase by applying HMMs. See for examples Tolba et al. [*85*], Khorsheed [*86*], and Al-Ma'adeed et al. [*52*] [*87*]. However, bypassing segmentation does not solve all Arabic OCR challenges.

Sari and Sellami reported a handwritten character segmentation algorithm for isolated words. The reported algorithm was based on topological rules, which were constructed during the feature extraction phase [*32*].

Some segmentation techniques divide the word into several segments where each segment could be a character, part of a character, or a group of more than one character. This might be done through morphological operations such as closing followed by opening [*88*]. A similar technique was used by Lorigo and Govindaraju [*89*] to over-segment the words into strokes and glyphs, then reduce the possible breakpoints using prior knowledge of letter shapes [*89*]. Elgammal and Ismail suggested a similar graph-based segmentation technique [*90*]. The suggested technique was based on the topological relation between the baseline and the line adjacency graph representation of the text, where the text is segmented into graph units representing sub-characters. Finally, a grammar-based tool is used to construct the characters from these units.

Kandil and El-Bialy [*91*] suggested a centreline independent segmentation technique based on upwards spikes that segment an image into isolated characters, diacritics, Hamzas, and sub-words or words.

Hadjar and Ingold presented a technique for extracting homogenous regions of complex structure in Arabic documents such as newspapers [*92*]. The authors have discussed other segmentation algorithms such as thread extraction, frame extraction, image text separation and text line extraction. Gouda and Rashwan [*93*] used discrete hidden Markov models to segment Arabic text into characters. A wavelet transform based segmentation algorithm was introduced by Broumandnia et al. in [*94*] where segmentation points were detected by the projection of horizontal edges and their location on baseline. Syiam et al. [*42*] described an Arabic OCR system that uses histogram clustering method for the segmentation of the Arabic words.

2.6 Features Extraction

The main objective of feature selection in recognition systems is to provide minimal and efficient representation for the original input data to maximize both the effectiveness and the efficiency of the recognition process, while minimizing the processing time and complexity. According to Cheriet et al. [*95*], feature extraction methods can be classified into three categories: geometric features, structural features, and feature space transformations methods. Examples of popular geometric features include moments, histograms, and direction features. Examples of structural features include registration, line element features, Fourier descriptors, and topological features. Examples of the transformation methods include principal component analysis (PCA) and linear discriminant analysis (LDA) [*95*].

Khorsheed and Clocksin used structural features for cursive Arabic words to recognize Arabic text using HMM [*70*]. The features used were the curvatures of word segments. The length of these segments was relative to other word segments' lengths, while the position was relative to the baseline and the description of curved word segments. The results of this method were used to train a HMM Model to perform the recognition. Jianying et al. features included loops, cusp distance and crossing distance [*96*]. Aburas et al. used different types of features which included structural features and statistical features. Some of these features were loops, endpoints, dots, branch-points, relative locations, height, sizes, pixel densities, histograms of chain code directions, moments and Fourier descriptors [*41*]. Ebrahimi et al. [*97*] used characteristic loci as part of their features. Al-Taani [*98*] suggested a feature extraction algorithm based on primary and secondary primitive features. Mahmoud in his digits recognition system [*99*] used unit features based on the digits. The extracted features were based on angle, distance, horizontal, and vertical-span features. Majumdar developed a feature extraction scheme based on the digital curvelet transform. The features included the curvelet coefficients of the image and its morphologically altered versions [*100*]. Ball used the character-based Word Model Recognizer (WMR) features [*24*]. The model consisted of 74 features. The features were described in details in [*101*]. Farah et al. proposed a system that used word-based structural features [*47*]. Gagne and Parizeau suggested sub-character based features based on the orientation and curvature of the strokes [*102*]. A feature fusion was proposed by Sun et al. [*103*]. They extracted two groups of feature vectors with the same sample and established the correlation criterion function between the two groups of feature vectors.

2.7 Classification and Hidden Markov Model (HMM)

Researchers are using different techniques to recognize printed Arabic text. These techniques includes statistical pattern recognition (Jain et al. [*104*]), structural pattern recognition (Gupta [*105*]), artificial neural networks (Al-Alawi [*106*], Al-Omaria and Al-Jarrah [*107*]), support vector machines (Bentouns [*44*], Pat and Ramakrishnan [*108*]), and multiple classifier methods (Wanas et al. [*109*] , Chang et al. [*110*]).

Most of the above recognition/classification techniques were developed to recognize isolated characters. When cursive text is considered, as a complete word or a complete string/line, a segmentation phase is needed to segment the image into isolated characters before using one of the above techniques. The segmentation process is generally believed to be error-prone (See Cheriet et al. [*95*]). This is one of the motivations for using HMM for the recognition of cursive Arabic script. No segmentation is needed for most of these cases, except to segment the page image into line images.

Initial results of a study related to using HMM to recognize handwriting Arabic text was presented by Zavorin and Eugene in [*111*]. The study was based on large-scale features, limited number of vocabulary that included 29 main Arabic characters. The training sets were machine printed images as templates.

Touj et al. proposed an approach for multi-writers Arabic handwritten recognition in [*112*]. The technique uses a hybrid planar Markov model to follow the horizontal and vertical variations of writing. The model is based on different segmentation levels: horizontal, natural and vertical. Experiments using planar Markov models for Arabic handwriting have shown promising results as reported in [*113*]. Their results varied from 47% to 67% for different fonts. However, when they considered selected 100 sub-words they reported an accuracy of more than 99% for those limited sub-words. HMM were also used for special purpose recognition including Indian numerals in Arabic script as reported in [*99*].

LaPre et al. used HMM based on BBN BYBLOS Speech Recognition System to recognize multi-font printed Arabic by modifying the feature extraction phase [*34*]. Khorsheed and Clocksin [*86*] [*114*] [*70*] used the HTK speech tool in Omni-font Arabic text recognition. The HTK is based on HMM. An accuracy rate of 65% was reported for a system that recognized Farsi handwritten words using discrete HMM by Dehghan et al. in [*43*]. A multi-font Arabic OCR system using Hough-transform for feature extraction and HMM for classifications with 96.8% accuracy in some cases was reported by Ben Amor and Ben Amra in [*45*]. Bazzi et al. reported a HMM system that could be used for recognition of English and Arabic printed text with accuracy reaching

95% for specific DARPA data [*46*]. Al-Ma'adeed et al. [*52*] [*87*] described a system for recognizing single handwritten Arabic words using the HMM approach.

2.8 Post-processing and Statistical Analysis

Sari and Sellami presented a contextual-based technique for correcting Arabic words generated by OCR systems in [*115*]. A rule-based system for correcting Arabic words operating only at the morpho-lexical level was used. An OCR system that uses linguistic information including affixes was proposed by Kanoun et al. in [*116*]. Borovikov et al. built a filter based post-OCR accuracy boost system [*117*]. The system combines different post-OCR correction filters, including a commercial spell-checker to improve the OCR results.

Statistical information of Arabic text could be used for post-processing. Few attempts have been carried out (See Section 1.5). These attempts include the work of Khedher and Abandah in [*118*], Elarian in [*119*], and Khorsheed in [*114*]. Statistical results were published in [*118*] for written Arabic syllables of length 1 to 8 letters. It also showed the percentages of these syllables. The analyzed text consisted of 252647 words and 1126420 characters. A second research work aiming to prepare an Arabic syllable dictionary for written Arabic to be used in OCR was introduced in [*119*]. The text used was taken from an Arabic newspaper. Al-Sulaiti [*10*] used a text of 842684 words for a similar purpose. The study provided syllables of length 1 to 17. The long syllables in the study were mainly due to typos in the used text. Several researchers have used the probability of Arabic letters in OCR based on HMM. Some of these researchers were Khorsheed [*114*], Bazzi et al. [*120*], and Schwartz et al. [*121*].

2.9 Commercial Arabic OCR Software

Several OCR software products with Arabic text recognition capabilities are available in the market. The following is a listing of some these products:

- Readiris™ Pro from I. R. I. S. is an OCR solution for converting paper documents into digital files. The software works for different languages. A Middle East version is available for Arabic, Farsi and Hebrew [*122*].
- VERUS™ Middle East Standard from NovoDynamics is designed to recognize Arabic, Farsi, Dari, and Pashto languages, including embedded English and French [*123*].
- Sakhr™ Automatic Reader from Sakhr is an OCR solution that addresses the Arabic language. It supports Arabic, Farsi, Pashto, Jawi, and Urdu. [*124*].
- OmniPage from Nuance Communications is an optical character recognition application that supports more than 25 languages including Arabic [*125*].

The data sheets of these software products claim a recognition rate reaching above 99%. However, no standard benchmarks were used for such claims. In one of the announcements of one of the softwares it says “Since version 11, Readiris™ increased recognition accuracy of 28%, especially on very complex documents and features a new algorithm for low resolution images”. This makes it unclear how the recognition rate exceeded 90%?

Independent researchers have evaluated earlier versions of some of these products by using different types of documents. The evaluation resulted in different percentages of recognition ranging from 10% to almost 100%. Several factors were affecting the recognition rates. Some of these factors were document quality, used fonts, and pre-trained fonts. Examples of OCR software evaluations can be found in Marton et al. [*126*] [*127*].

2.10 Summary

The first question that might arise is how to compare the performance of the reported OCR systems? The reported systems used different datasets for different purposes and different applications. Systems designed to recognize only numerical digits consisting of ten isolated shapes cannot be compared to systems designed to recognize isolated or cursive letters consisting of more than a hundred shapes. However, comparisons among systems addressing the same datasets do exist, as reported in [*51*] [*128*] for Tunisian towns.

OCR systems usually handle special purpose data or open vocabulary data. Special purpose data could be classified into several categories: numerals, postal addresses, literal amounts, and isolated letters in forms. Each of which has its own applications. Each type of these categories needs it own specified datasets for training and testing.

By reviewing the available databases for Arabic text recognition, it is clear there is an urgent need for publicly available databases to be used as benchmarks. A trusted database benchmark should have its transcription (ground truth information) 100% correct and accurate. It is not clear if the databases reported in literature contain accurate statistical distribution for the different shapes of Arabic characters. In some cases, some characters are appear 50 times more compared to other characters (See [*65*] as an example). Moreover, in the case of handwriting, if a database is targeted, it will be very hard to require writers to write long text, say one page or more. Even if we are able to collect two handwritten pages or more per writer, some of the characters may not be present in the text with adequate frequency. We have noticed that none of the available handwritten databases claimed that it covers all the basic shapes of Arabic letters. One of the objectives of this research work is to tackle this research gap. We hope to contribute towards providing open vocabulary Arabic printed datasets with different fonts for researchers.

A wide ranges of different feature extraction schemes were used in the literature. In some research works tens of features of different types were extracted. We have noticed that the trends were to use different types of features in the same recognition process to represent the addressed characters. Other than the complexity and the over-head of using different types of features, the reported accuracies did not meet the expectations. This research aims to introduce a simple feature extraction technique that represents the images and keeps the uniqueness of different characters in the image to help in accurate classifications. The suggested feature extracting technique will be used to recognize printed Arabic text of different fonts based on the built database. To avoid explicit segmentation of text, which has proven to be error-prone to erroneously segmented characters, we will use HMM for classification as it does not need explicit segmentation.

Researchers working on Arabic text recognition are accustomed to using Arabic letters as the basic unit of classification. This research work will explore using the shape of the letter as the basic unit of classification. In the former method, all different shapes of an Arabic letter are considered as one class. In our method, an Arabic letter with four basic shapes is given four different classes: a class for each shape.

Arabic multi-font recognition is still new research area. This research gap will be also tackled through this research.

Few research efforts have been put towards post-processing of Arabic OCR. This research work will contribute in this area and new efficient techniques will be proposed.

The next chapter introduces the statistical analysis carried out to better understand the nature of Arabic text. It also covers the preparation of the printed Arabic datasets that will be used through this research work.

Chapter 3. Statistical Analysis and Data Preparation

3.1 Introduction

In order to better understand the features of the Arabic language, a statistical and analytical analysis of an Arabic text was carried out. The results of this analysis could be extremely useful for Arabic OCR research. The statistics are useful in choosing Arabic text for a database benchmark to ensure fair representation of standard classical Arabic. They also construct the language model that will be used in the classification phase. The classification phase when including a language model needs such statistics. The post-processing phase could also benefit from these statistics. It is worth mentioning that standard classical Arabic was used in writing Islamic culture and phylosophy, Islamic supplementary material, Islamic believes, History, Jurisprudence, etc. The modern standard Arabic is a form of classical Arabic that is being used and understood in all countries of the Arab world.

Research on Arabic OCR is not as advanced as research on Latin OCR. One of the reasons behind this is the lack of public benchmark databases. Most of the current research on printed Arabic OCR is carried out on private datasets. Even when a researcher could get a colleague's database through personal communications, it is very hard to ensure that the provided ground truth information for such database is accurate. There is an immediate need to have public databases for printed Arabic text and make them available publicly for researchers. One objective of this research work is to prepare a database to be used throughout this research work and to make it public for the scientific research community [*129*].

This chapter presents a summary of the statistical analysis that has been carried out and describes the new printed Arabic text database sets used in this research work in addition to the minimal dataset that we have introduced to cover all the possible basic shapes of Arabic alphabets. The chapter is organized as follows. Section 3.2 describes the text used for statistics. Section 3.3 defines the terminology used in this chapter. A summary of the statistics is presented in Section 3.4. The detailed statistics are provided in the enclosed CD-ROM (See Appendix A). Section 3.5 presents the source of the selected data and describes the two prepared datasets. Section 3.6 presents the statistics of the characters in each dataset. Data labelling with ground truth information is presented in Section 3.7. The minimal Arabic script is described in Section 3.8. Section 3.9 presents the developed tool for coding and decoding the data used. Section 3.10 shows the difference between the synthesized images and the scanned images. The status of the webpage, where the datasets are published, is presented in Section 3.11. A summary of this chapter is presented in Section 3.12.

3.2 Text Used

In order to statistically analyze Arabic text, two Arabic books have been chosen. The chosen books of *Saheh Al-Bukhari* and *Saheh Muslem* [*130*] [*131*] represent standard classical Arabic. The standard classical Arabic is the language that has been used by all scholars. These two books were chosen because they are valuable historical manuscripts that represent classic Arabic literature and are valued by hundreds of millions of people around the Globe. A second reason was that they represent a variety of Arabic alphabets open vocabulary. Many old scanned Arabic books written in standard classical Arabic are not available in digital text format. The text under consideration included 4,405,318 characters representing 1,095,274 words. The count of unique words is 50,367.

3.3 Definitions

The following are the list of terms used in the statistical analysis:

- **Syllable**: connected letters in one word.
- **Isolated**: a letter is isolated if it is not connected to the previous letter or the following letter, (i.e., it is a standalone syllable). For example, the Arabic word (أن) has two syllables each is an isolated letter. A letter might be isolated in one syllable and not isolated in a different syllable.
- **Connected**: A letter is connected if it is connected to the previous letter, to the next letter, or to both letters. The Arabic word (مرتبطا) has two syllables. The first one is (مر) and the second one is (تبطا). The letters of the syllables are connected.
- **First letter**: The first letter in a syllable.
- **Last letter**: The last letter in a syllable.
- **Syllable length**: Number of letters in the syllable.
- **N-Gram**: A subsequence of *n* letters from a given word. The size of N-Gram is *n*. If *n* is one then it is called **unigram**, if *n* is two it is called **bigram**, and if *n* is three it is called **trigram**.

3.4 Statistics

The results of the analysis are tables showing the frequencies of Arabic letters, shapes and syllables in Arabic. These results include:

- Frequency of each Arabic letter according to letter shapes.
- Frequency of each Arabic letter in each syllable.
- Frequencies of bigrams (a letter and its following letter) in each syllable.
- Percentage of usage of Arabic letters and syllables.

3.4.1 Statistics for shapes of letters

Table 3-1 shows the frequencies of each letter with its appearances in different shapes in *Al-Bukhari* Book [*130*] .

Arabic letters may have up to 4 shapes depending on their classes (See Section 1.2). The Arabic letter *Hamza* (ء) has only one shape. It is always not connected (Stand-alone). Other Arabic letters may appear in only two shapes like the letter *Daal* (د) and *Raa* (ر). This type of letters with 2 shapes appears either stand-alone or connected from right (terminal). The third class of Arabic letters has 4 shapes. The letter could be the start of a word and connected from left (initial). It could be in the middle of a word and connected from both sides (Medial). It could also be in a terminal position connected from right. The fourth case is when it is not connected (stand-alone).

Table 3-2 shows the frequencies of letters according to their shapes in *Muslem's* book [*131*]. The frequencies of shapes of letters in both books are shown in Table 3-3.

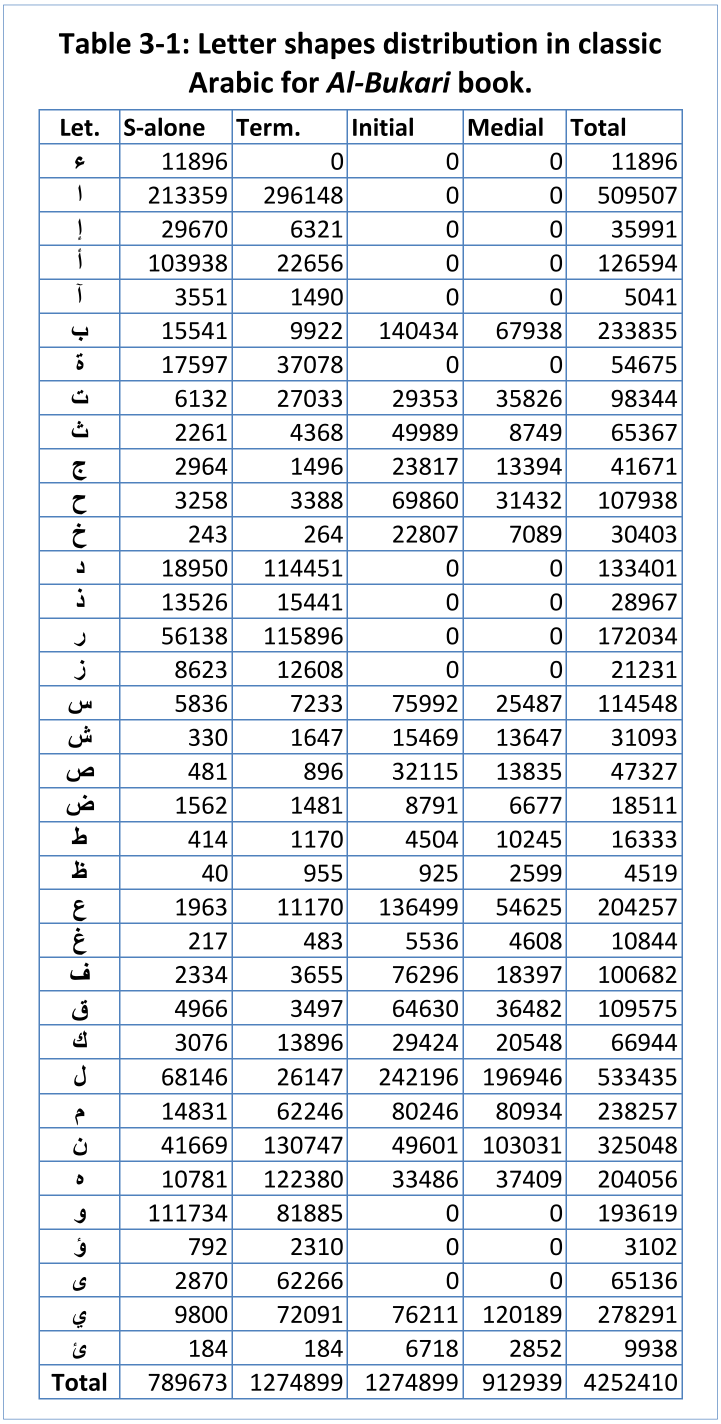


Table 3-1: Letter shapes distribution in classic Arabic for *Al-Bukari* book.

Let.	S-alone	Term.	Initial	Medial	Total
ء	11896	0	0	0	11896
ا	213359	296148	0	0	509507
إ	29670	6321	0	0	35991
أ	103938	22656	0	0	126594
آ	3551	1490	0	0	5041
ب	15541	9922	140434	67938	233835
ة	17597	37078	0	0	54675
ت	6132	27033	29353	35826	98344
ث	2261	4368	49989	8749	65367
ج	2964	1496	23817	13394	41671
ح	3258	3388	69860	31432	107938
خ	243	264	22807	7089	30403
د	18950	114451	0	0	133401
ذ	13526	15441	0	0	28967
ر	56138	115896	0	0	172034
ز	8623	12608	0	0	21231
س	5836	7233	75992	25487	114548
ش	330	1647	15469	13647	31093
ص	481	896	32115	13835	47327
ض	1562	1481	8791	6677	18511
ط	414	1170	4504	10245	16333
ظ	40	955	925	2599	4519
ع	1963	11170	136499	54625	204257
غ	217	483	5536	4608	10844
ف	2334	3655	76296	18397	100682
ق	4966	3497	64630	36482	109575
ك	3076	13896	29424	20548	66944
ل	68146	26147	242196	196946	533435
م	14831	62246	80246	80934	238257
ن	41669	130747	49601	103031	325048
ه	10781	122380	33486	37409	204056
و	111734	81885	0	0	193619
ؤ	792	2310	0	0	3102
ى	2870	62266	0	0	65136
ي	9800	72091	76211	120189	278291
ئ	184	184	6718	2852	9938
Total	789673	1274899	1274899	912939	4252410

Table 3-2: Letter shapes distribution in classic Arabic for *Muslim* book.

Let.	S-alone	Term.	Initial	Medial	Total
ء	4882	0	0	0	4882
ا	93283	130023	0	0	223306
إ	12574	3145	0	0	15719
أ	48367	9591	0	0	57958
آ	1292	672	0	0	1964
ب	6220	4792	72910	30187	114109
ة	7759	17666	0	0	25425
ت	2492	11262	10148	14771	38673
ث	1231	2747	26245	5017	35240
ج	1214	804	10618	5683	18319
ح	2434	1515	36727	16800	57476
خ	140	114	10513	2931	13698
د	8844	57376	0	0	66220
ذ	5625	7008	0	0	12633
ر	24844	55294	0	0	80138
ز	4648	5493	0	0	10141
س	2150	3312	36342	10328	52132
ش	148	801	7531	5976	14456
ص	244	372	13953	5710	20279
ض	664	556	2124	2976	6320
ط	137	441	1895	4349	6822
ظ	21	784	372	1064	2241
ع	784	5315	61392	23589	91080
غ	84	190	2415	1900	4589
ف	899	1264	31638	8346	42147
ق	2737	1366	28335	15427	47865
ك	1361	5757	13278	8805	29201
ل	30787	10964	102656	85995	230402
م	6196	26451	34919	37982	105548
ن	17620	66300	22569	46892	153381
ه	4695	52532	15688	15670	88585
و	52251	37023	0	0	89274
ؤ	273	938	0	0	1211
ى	1199	25422	0	0	26621
ي	4442	33522	35974	56828	130766
ئ	91	90	2660	1148	3989
Total	352632	580902	580902	408374	1922810

Table 3-3: Letter shapes distribution in classic Arabic For *Al-Bukari* and *Muslim*.

Let.	S-alone	Term.	Initial	Medial	Total
ء	11896				11896
ا	213359	296148			509507
إ	29670	6321			35991
أ	103938	22656			126594
آ	3551	1490			5041
ب	15541	9922	140434	67938	233835
ة	17597	37078			54675
ت	6132	27033	29353	35826	98344
ث	2261	4368	49989	8749	65367
ج	2964	1496	23817	13394	41671
ح	3258	3388	69860	31432	107938
خ	243	264	22807	7089	30403
د	18950	114451			133401
ذ	13526	15441			28967
ر	56138	115896			172034
ز	8623	12608			21231
س	5836	7233	75992	25487	114548
ش	330	1647	15469	13647	31093
ص	481	896	32115	13835	47327
ض	1562	1481	8791	6677	18511
ط	414	1170	4504	10245	16333
ظ	40	955	925	2599	4519
ع	1963	11170	136499	54625	204257
غ	217	483	5536	4608	10844
ف	2334	3655	76296	18397	100682
ق	4966	3497	64630	36482	109575
ك	3076	13896	29424	20548	66944
ل	68146	26147	242196	196946	533435
م	14831	62246	80246	80934	238257
ن	41669	130747	49601	103031	325048
ه	10781	122380	33486	37409	204056
و	111734	81885			193619
ؤ	792	2310			3102
ى	2870	62266			65136
ي	9800	72091	76211	120189	278291
ئ	184	184	6718	2852	9938
Total	789673	1274899	1274899	912939	4252410

3.4.2 Statistics of Syllables

The total number of syllables in the analyzed text is 2,217,178 with 18,170 unique syllables. The total number of characters is 4,405,318. The text under consideration included 1,095,274 words with 50,367 unique words.

The results are available in softcopy format as they are presented in more than 300 pages (See Appendix A). However, the following several tables display some of the results. The results could be used efficiently in Arabic text recognition in several phases, including the recognition phase using HHM and the post-processing phase.

Table 3-4 shows the first 350 highest frequencies syllables of the 18170 unique syllables. It is worth noting that 10% of the total syllables are for the character *Alef* (ا).

Table 3-4: The 350 most frequent Arabic syllables.

Syl.	%	Syl.	%	Syl.	%	Syl.	%	Syl.	%	Syl.	%
ا	9.6243	و	5.0401	أ	4.6884	ل	3.0741	لله	2.6366	ر	2.5324
بن	2.4033	حد	1.9598	ن	1.8796	قا	1.8777	عن	1.794	ثنا	1.432
)	1.3445	(	1.3443	إ	1.3385	عليه	1.054	صلى	0.9907	سلم	0.9786
د	0.8547	ة	0.7936	من	0.7781	لا	0.7274	سو	0.7138	ب	0.7009
بي	0.6725	م	0.6691	]	0.6575	[	0.6575	ما	0.6419	با	0.618
ذ	0.6102	عبد	0.5851	في	0.5614	ء	0.5367	نا	0.5082	ه	0.4862
لى	0.4743	فقا	0.4485	ي	0.442	بو	0.4281	لأ	0.4192	لنبي	0.4033
ز	0.389	1	0.3751	خبر	0.3578	2	0.3426	كا	0.3291	على	0.3285
عمر	0.3153	لر	0.3111	لك	0.2902	ثم	0.2846	ني	0.283	محمد	0.279
ت	0.2767	فأ	0.2725	لو	0.272	له	0.2646	س	0.2632	ير	0.2598
يا	0.2569	مر	0.2566	يو	0.2552	يد	0.2516	ثني	0.2469	5	0.244
6	0.2431	بر	0.2419	3	0.2417	4	0.2416	تعا	0.234	عا	0.233
ق	0.2239	هر	0.215	ضى	0.2082	7	0.2014	}	0.2008	9	0.1981
{	0.1961	لت	0.192	8	0.1907	لنا	0.1907	0	0.1891	يحيى	0.1861
نه	0.1833	جا	0.1828	حتى	0.1806	لم	0.1797	سعيد	0.1746	كر	0.1711
قد	0.1709	فا	0.1697	لد	0.1643	ها	0.1605	آ	0.1602	يقو	0.1556
بكر	0.1528	فإ	0.1525	جل	0.152	هو	0.1499	هذ	0.1472	ح	0.1469
نس	0.1452	لز	0.1446	خر	0.1432	مو	0.1432	ك	0.1387	به	0.1352
عنه	0.1348	سمعت	0.134	ج	0.1338	ى	0.1294	لذ	0.1266	قو	0.1235
عر	0.1167	لإ	0.1166	فر	0.116	عبا	0.1154	جر	0.113	سا	0.1116
مة	0.1106	لي	0.1096	هيم	0.1094	حر	0.1083	ية	0.1076	فع	0.1068
ين	0.1064	ئشة	0.1056	حمن	0.1056	ف	0.1053	ث	0.1019	هل	0.1013
علي	0.1005	صا	0.0972	هم	0.0964	يث	0.0943	قر	0.0937	يز	0.0935
حا	0.0888	سفيا	0.0884	ع	0.0884	شعبة	0.0876	سحا	0.0872	خا	0.0869
بيه	0.0867	عبيد	0.0866	عند	0.0866	عو	0.0825	سما	0.0812	هب	0.0776
كل	0.0767	نت	0.0763	قلت	0.0753	سى	0.0752	غير	0.0734	تر	0.0732
شيبة	0.0726	بين	0.072	بعد	0.0714	ض	0.0705	عنهما	0.0687	فلما	0.0674
شها	0.0665	ته	0.0663	كم	0.0654	فيه	0.0638	مع	0.0634	هما	0.0631
معا	0.0631	هشا	0.0622	فلا	0.0621	خذ	0.0617	سلمة	0.0615	عيل	0.0615
لما	0.0608	نز	0.0608	لآ	0.06	بها	0.0593	فو	0.0584	فقلت	0.058
سعد	0.0566	سمع	0.056	عد	0.0556	بهذ	0.0548	لصلا	0.0541	خل	0.0539
نها	0.0537	سنا	0.0533	ثا	0.0527	شر	0.0525	قبل	0.0522	ثلا	0.052
يت	0.0519	لها	0.0517	عنها	0.0497	جد	0.0497	هير	0.0494	قتيبة	0.0493
نو	0.0493	شي	0.0489	سليما	0.0488	منه	0.048	هد	0.0479	لمثنى	0.0476
سأ	0.0472	كذ	0.0471	قتا	0.0467	يعني	0.0467	تو	0.0466	يأ	0.046
عة	0.0458	جعفر	0.0457	طا	0.0457	بأ	0.0455	شا	0.045	نصا	0.0449
بة	0.0447	بد	0.0442	كنت	0.044	صلا	0.0437	تا	0.0433	ليه	0.0429
بير	0.0425	فيها	0.0422	نعم	0.0422	لهم	0.0421	عمش	0.0419	لجنة	0.0419
خير	0.0417	فد	0.0416	لمد	0.0415	لحد	0.0414	عثما	0.0414	فذ	0.0409
لحا	0.0404	ليس	0.0401	جلا	0.04	لقر	0.0397	سر	0.0396	صد	0.0396
شيئا	0.039	بت	0.0388	ينة	0.0384	حما	0.0383	ينا	0.0383	نما	0.0382
حين	0.0382	بني	0.0381	صحا	0.0373	فلم	0.0371	عطا	0.0371	لقا	0.0371
قة	0.0367	جو	0.0367	حميد	0.0367	للهم	0.0363	لمر	0.0362	ؤ	0.0357
مه	0.0353	لقد	0.0352	معمر	0.0352	لسا	0.035	منها	0.035	كما	0.0347
كنا	0.0342	منا	0.0341	يب	0.0341	لليث	0.034	مثل	0.0335	كو	0.0334
لح	0.0333	يج	0.0332	حمد	0.0332	لمؤ	0.0331	تي	0.033	بيد	0.0325
حب	0.0324	بنت	0.0322	سف	0.032	يذ	0.032	مي	0.0318	نك	0.0316
نمير	0.0316	هي	0.0311	نسا	0.0309	لقو	0.0308	سم	0.0306	تى	0.0295
معه	0.0295	للفظ	0.0293	نة	0.0293	بشا	0.0292	فما	0.0292	تم	0.029
بما	0.0285	بنا	0.0285	بشر	0.0285	جميعا	0.0284	ليد	0.0284	صو	0.0282
سلا	0.0281	لمسجد	0.028	جع	0.0279	عشر	0.0275	لعز	0.0273	منصو	0.0272
لكم	0.0272	كلا	0.027	شد	0.0267	لتي	0.0266	بك	0.0258	حو	0.0258
فيقو	0.0257	مسلم	0.0257	نحو	0.0257	صم	0.0256	كثير	0.0256	لحسن	0.0254
ثو	0.0252	ضي	0.0252	لقيا	0.0252	يحد	0.0251	بيع	0.0243	سه	0.0243
يقا	0.0242	لسما	0.0239	عليها	0.0238	يصلي	0.0235	عز	0.0234	تأ	0.0233
ثة	0.0232	لبيت	0.0232	لعا	0.0231	منهم	0.0231	نهم	0.0231	تقو	0.0231
تد	0.023	علم	0.023	لليل	0.0229	يه	0.0229	لسلا	0.0227	فكا	0.0226

Table 3-5 shows the Frequencies and lengths of syllables. 90% of the syllables have length of 3 or less, 98% of syllables are of length 4 or less.

Table 3-5: Frequencies and lengths of syllables.

Syll. length	count	frequency	%
1	57	942097	42.49079
2	537	638359	28.79149
3	3192	418885	18.89270
4	6778	169938	7.664604
5	4896	39026	1.760164
6	2118	7361	0.331998
7	522	1322	0.059625
8	62	162	0.007306
9	8	28	0.001262
Total	18170	2217178	100

Longer syllables have low probabilities and low frequencies. There are only 8 syllables of length 9 (shown in Table 3-6).

Table 3-6: All syllables of length 9 with percentages.

Syllable	%	Syllable	%	Syllable	%
لمستضعفين	0.00086	فليقطعهما	0.00009	قسطنطينية	0.00009
ليثنينهما	0.00005	فلتقتلنهم	0.00005	يستبينهما	0.00005
مستقبليها	0.00005	فليلتمسها	0.00005		

As the length of syllables decreases the number of different syllables increases. Table 3-7 shows all syllables of length 8 with their percentages.

Table 3-7: All syllables of length 8 with percentages.

Syllable	%	Syllable	%	Syllable	%	Syllable	%	Syllable	%
مستخلفكم	0.00005	مستخلفين	0.00005	ستطعمتها	0.00005	فليحملها	0.00005	للمطففين	0.00005
ملتصقتين	0.00005	فسقيتهما	0.00005	لمجنبتين	0.00005	لمستقبلة	0.00005	لمطمئنين	0.00005
فهيجتهما	0.00005	بحبيبتيه	0.00005	فليحلبها	0.00005	تستطيعها	0.00005	فتستقبله	0.00005
فجعلتهما	0.00005	تخفيفهما	0.00005	يستحسنها	0.00005	فغمستهما	0.00005	يستكملها	0.00005
فقبضتهما	0.00005	لقطعتكما	0.00005	ليبتليكم	0.00005	فليمسكها	0.00005	فليمتهما	0.00005
يستعملها	0.00005	فمنعنيها	0.00005	فجمعتهما	0.00005	لمقتسمين	0.00009	فلبستهما	0.00009
يستقبلكم	0.00009	ففظعتهما	0.00009	ليخلعهما	0.00009	فليجعلها	0.00009	فليستجمر	0.00009
فليستغفر	0.00009	فليستنشق	0.00009	فليتحلله	0.00009	فليمنحها	0.00009	يستلمهما	0.00009
فتبيعنيه	0.00009	لمتخلفين	0.00009	بمغنيتين	0.00009	مستضعفين	0.00009	ستقبلهما	0.00009
تسليمتين	0.00009	تصلينهما	0.00014	لخليفتين	0.00014	لجهنميين	0.00014	لينعلهما	0.00014
لمتشبهين	0.00018	فليستنثر	0.00018	تستعملني	0.00018	تستعينها	0.00018	للمحلقين	0.00027
لمتكلفين	0.00027	فليستعفف	0.00027	بسيفيهما	0.00032	فنفختهما	0.00036	ليقطعهما	0.00036
فليطلقها	0.00045	للمسلمين	0.00086						

In Table 3-8 frequencies of each letter in different lengths of syllables are presented.

Table 3-9 presents frequencies of letters appearing as a first letter in the syllables with different lengths. Letters of class 2 may not be in the first position of a connected syllable. Hence, none of them appears in this table in the first position.

Frequencies of bigrams of letters in the first and second positions of syllables are shown in Table 3-10. Table 3-11 also presents bigram frequencies of letters in the second and third positions of syllables.

Table 3-8: Frequency of letters in their syllables.

Letter	Its frequency in syllables of specified length									Total	%
	1	2	3	4	5	6	7	8	9		
ء	11900									11900	0.27013
ء	11900									11900	0.27013
آ	3552	1459	22	6	3					5042	0.11445
أ	103951	20129	2057	430	38	2				126607	2.87396
ؤ	792	1257	995	55	3					3102	0.07041
إ	29677	6277	43	1						35998	0.81715
ئ	6903	1374	1289	298	71	4	2			9941	0.22566
ا	213387	139833	113405	29471	10664	2251	475	75	6	509567	11.56709
ب	155995	46774	23260	6578	1199	44	8			233858	5.30854
ة	17596	10867	8824	11377	5350	625	40	1	2	54682	1.24127
ت	35488	33473	15133	11861	2205	203	1			98364	2.23285
ث	52252	7909	3554	1540	77	37	4			65373	1.48396
ج	26785	10529	3320	968	58	18				41678	0.94608
ح	73129	28380	4373	1729	288	50	2			107951	2.45047
خ	23053	5884	1186	248	31	4				30406	0.69021
د	18951	65359	30893	16728	1401	76	7			133415	3.02850
ذ	13529	11690	3202	426	102	18	7			28974	0.65771
ر	56148	53029	44166	15879	2392	420	20	8		172062	3.90578
ز	8625	9087	2960	499	52	11				21234	0.48201
س	81833	20162	9913	2325	273	48	1			114555	2.60038
ش	15804	11727	3060	466	36	3	2			31098	0.70592
ص	32597	10323	3605	692	101	11				47329	1.07436
ض	10357	4336	3008	697	107	9	2			18516	0.42031
ط	4917	7070	3082	1108	155	7				16339	0.37089
ظ	965	1492	958	944	143	18				4520	0.10260
ع	138472	46144	14845	3947	684	181	9			204282	4.63717
غ	5753	2932	1604	515	36	6				10846	0.24620
ف	78655	12457	6785	2098	584	86	33	6		100704	2.28596
ق	69603	30605	6591	2352	350	88	1	2		109592	2.48772
ك	32502	23626	6168	3832	710	101	14			66953	1.51982
ل	310393	184785	26657	9909	1499	253	21			533517	12.11075
م	95101	68082	54796	14581	4783	799	148	8	1	238299	5.40935
ن	91279	198498	21691	9593	2906	638	401	56	19	325081	7.37929
ه	44267	36817	85099	32520	4291	956	127	9		204086	4.63272
و	111749	56242	15456	7272	2261	588	73			193641	4.39562
ى	2869	18616	35409	6550	1659	39				65142	1.47871
ي	86024	87855	79313	20342	3387	1279	114	25		278339	6.31825
أخرى										152325	3.45775

Table 3-9: Frequency of letters as a first letter in the syllables.

	Its Frequency as a 1st letter in syllables of specified length									
1st Letter	**1 (Isolated)**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Total**
ء	11900									11900
آ	3552									3552
أ	103951									103951
ؤ	792									792
إ	29677									29677
ئ	184	2912	3602	160	45					6903
ا	213387									213387
ب	15540	106216	24975	6286	2456	483	29	10		155995
ة	17596									17596
ت	6134	9610	13122	4191	1885	471	60	15		35488
ث	2260	9600	39515	743	121	13				52252
ج	2966	14302	5296	2993	1160	64	4			26785
ح	3256	51019	14474	3682	624	67	7			73129
خ	243	8895	11693	1487	636	77	22			23053
د	18951									18951
ذ	13529									13529
ر	56148									56148
ز	8625									8625
س	5836	25053	34475	13131	2897	328	110	3		81833
ش	330	4793	4529	6015	121	15	1			15804
ص	481	4886	25817	1097	270	40	6			32597
ض	1563	7559	793	349	83	10				10357
ط	413	2025	1374	916	161	28				4917
ظ	40	225	529	146	17	8				965
ع	1961	53599	46091	31891	4676	231	23			138472
غ	217	1365	3346	612	180	32	1			5753
ف	2334	35292	24425	11738	3133	1340	331	58	4	78655
ق	4964	52950	7924	1920	1773	68	2		2	69603
ك	3075	17678	8273	2661	743	51	21			32502
ل	68159	87502	95723	44072	12144	2181	528	64	20	310393
م	14835	47959	14292	15878	1694	376	61	5	1	95101
ن	41675	31471	12424	4324	1076	281	28			91279
ه	10779	23695	8722	948	114	9				44267
و	111749									111749
ى	2869									2869
ي	9801	39753	17471	14698	3017	1188	88	7	1	86024

Table 3-10: frequency of bigrams (1st & 2nd).

1st Let	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ى	ي	Isolated	Total
ئ						8	299	373	141	4	6	39		264	89	180	95	10	2347	1	144	113		45	9	165	32	323	488	642	104	412	6	2	378	184	6903
ا																																				213387	213387
ب	58	1009	6	298	115	13703	258	991	1518	168	170	693	275	980	404	5364	80	681	1683	487	91	467	44	5005	355	140	789	4933	1709	2050	56482	6336	9491	175	23447	15540	155995
ة																																				17596	17596
ت		517	176		10	961	968	14	360	30	474	1174	601	510	150	1623	349	1019	324	681	171	333	49	6038	248	542	1449	1159	1000	1532	919	2091	1033	655	2194	6134	35488
ث					1	1169	124	515	294		1		2	68		298								25	4	2	132	156	1251	6704	37838	522	559	10	317	2260	52252
ج		9	4		196	4052	1209	257	738	17	3	143		1101	86	2506	228	81	47					2750		53		60	5186	1512	892	1541	813	24	311	2966	26785
ح						1968	1978	157	4300	50	1109			43452	283	2402	187	1229	137	419	180	58	37			832	548	272	986	5984	306	127	573	92	2207	3256	73129
خ						1927	8282	14	474	27				220	1367	3176	85	87	265	242	81	387		1		248		1	3119	505	53	8	394	39	1808	243	23053
د																																				18951	18951
ذ																																				13529	13529
ر																																				56148	56148
ز																																				8625	8625
س	2	1047	20		170	2475	1256	59	2519		426	2029	67	247		879		5				188		5270		2990	415	276	26121	7819	1736	1423	15827	1668	1063	5836	81833
ش		193	4		240	997	372	37	475	1	294	60	15	592		1164	5	4				68		3019	33	255	310	232	6	333	49	2465	154	8	4089	330	15804
ص						2155	828	54	10			1014	63	877		193						65		133	114	523	2	11	23704	620	357	162	626	41	564	481	32597
ض		360	17		20	329	44	94	153		8	216	21	2		399						95		801	9	6		21	91	75	50	157	420	4616	790	1563	10357
ط		39	2		16	1013	149	35	16			12				226		18	5					512	10	95	1	17	1002	268	34	179	419	10	426	413	4917
ظ					3	60	10	3								18								5		44			190	6	200	369	2		15	40	965
ع						5165	17560	1016	1038	938	323			1233	338	2587	518	164	697	224	56	1510	317	1		307	982	406	37005	9684	48249	686	1829	135	3543	1961	138472
غ						280	143	3	174	22				231	4	270	373	372	51	24	198	67				359			442	100	488	26	42		1867	217	5753
ف	65	6041	6	3381	46	3762	677	336	2041	68	1385	672	722	923	906	2573	206	1295	315	907	1111	437	79	3834	267	341	13300	1353	5608	1495	1344	1099	1294	181	18251	2334	78655
ق		15				41631	2303	814	3405	2		47		3789	43	2078	54	214	18	287	444	534	24	427		209	10	55	3023	305	187	362	2738	31	1590	4964	69603
ك	3	412		12	6	7297	958	120	1224	1033	4	52	9	57	1045	3793	22	279	61	10	21	8	2	1333	5	554	47	24	3876	2404	2196	348	741	14	1457	3075	32502
ل	1330	9294	40	2586	216	16127	3814	435	7385	1139	3433	7750	2555	3642	2806	6898	3207	4280	2691	3576	482	1261	388	5693	1062	2082	5632	9618	63348	17858	16602	8796	6031	10516	9661	68159	310393
م	1	82	230		234	14233	157	2453	1375	1573	679	7037	425	345	39	5690	196	3486	323	282	602	288	40	5954	238	112	488	1166	1445	718	23598	1475	3176	142	1984	14835	95101
ن		91	28		68	11267	1235	650	2465	74	370	1174	220	184	156	185	1349	4510	228	1857	119	627	392	1558	58	1713	503	1351	26	1696	274	6614	1092	199	7271	41675	91279
ه			251			3559	2136	52	92		110			1063	3264	4767	61		1566	3	5	103					35	300	3779	3612	623	197	3323	58	4527	10779	44267
و																																				111749	111749
ى																																				2869	2869
ي		1020	473		33	5695	2014	2385	3276	2763	1734	6268	909	5579	710	5760	2072	2428	969	1323	631	461	120	3740	520	885	5930	1892	1380	2160	5917	1422	5659		95	9801	86024

Table 3-11: Frequency of bigrams (2nd & 3rd).

2ndLet	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ى	ي	Isolated	Total
آ																																				1459	1459
أ																																				20129	20129
ؤ																																				1257	1257
إ																																				6277	6277
ئ						19		31	402					17	219	76	1	40			8							108	168		161	2	10		14	98	1374
ا																																				139833	139833
ب		111	1	4	96	4502	106	291	806	96	36	678	90	13281	45	9453	70	362	139	126	112	187	3	1216	133		362	439	2226	7	431	618	441	15	5805	4486	46774
ة																																				10867	10867
ت		399	6		21	1969	1082	78	141	47	255	723	261	267	41	937	187	359	167	283	47	224	71	518	273	385	782	820	1217	1459	907	1909	759	4100	2830	9949	33473
ث	1					488	63	141		2			3	10		512								19	4		202	8	1647	1094	128	428	226	49	711	2173	7909
ج		19	3		131	1770	383	221	117	7	21	141	3	817	38	1002	217	45	17	2				915		21		22	558	729	1245	355	326	14	376	1014	10529
ح						4003	671	119	391	47	986	8		1860	98	1233	86	876	115	159	142	63	5			128	427	497	981	7451	643	98	972	79	4810	1432	28380
خ						352	289		169	3				423	322	1123	102	38	85	77	66	701				155			682	373	119		151		598	56	5884
د																																				65359	65359
ذ																																				11690	11690
ر																																				53029	53029
ز																																				9087	9087
س	5	816	74		30	2464	704	62	1361		512	197	106	513		860		9				149		919	3	301	210	544	2242	1610	862	323	503	56	1001	3726	20162
ش		57	14			2634	122	2356	217		171	21	19	142		2005	2		6			46		464	47	219	214	115	1	582	27	389	127	62	1282	386	11727
ص						1794	787	86	59			136	49	750		858				24		9		169	42	560	17	8	2994	275	264	50	434	52	740	166	10323
ض		3			1	734	266	86	54		26	357	9	9		714						48		197	16	9			562	68	8	19	58	222	513	357	4336
ط		76	19		6	1508	224	23	33		67	43	1	1		630		38	18			13		577	2	228	11	14	900	62	220	173	519	178	923	563	7070
ظ					1	79	9	64								392								25		22		4	160	149	84	322	12	2	120	47	1492
ع						7869	3167	580	1461	680	245			3387	231	1295	653	134	529	452	1043	375	124			1189	727	208	2915	2882	2109	1690	633	35	5641	5890	46144
غ						218	45	12	100	7				183	5	281	108	234	65	10	195	52				187			329	40	258	1	66	6	494	36	2932
ف		36	20		38	1026		405	682	15	252	48	58	53	19	1323	87	1190	18	630	419	212	156	585		27	324	20	274	6	119	148	306	27	2653	1281	12457
ق		1				12010	1081	56	771	7		9		1637	57	1805	14	261	39	316	290	426	16	316		75	6	18	2055	381	124	271	5018	138	2254	1153	30605
ك	1	89				1152	344	644	781	157		91		29	452	4176	3	202	55	4			1	228		415		14	952	1867	1346	76	865	26	641	9015	23626
ل	13	185	19	38	9	7641	887	671	2637	82	67	881	33	283	106	75	54	344	36	123	6	78	40	402	493	1329	1419	930	136	27334	912	61098	1275	29486	34037	11626	184785
م	1	28	735	1	7	10376	190	552	632	1853	243	374	162	1753	10	8988	450	2786	1506	264	101	268	26	5808	555	139	445	532	1698	117	4879	650	764	113	3564	17512	68082
ن	1	30	2			41704	9845	1346	2444	16	238	412	213	2363	175	26	746	646	91	906	331	107	496	873	6	697	392	1137	15	201	134	9244	659	306	8874	113822	198498
ه			15			6930	110	14	64		118			1311	1328	917	139	5	6		14	3					5	49	972	3722	550	401	648	439	833	18224	36817
و																																				56242	56242
ى																																				18616	18616
ي		207	87		949	2163	2885	986	1811	535	83	254	179	1804	56	5485	41	2344	168	229	234	121	20	1614	30	699	1048	681	2973	4387	6161	6834	684	4	599	41500	87855

3.5 Arabic Printed Datasets

We introduce here two printed Arabic datasets: (PATS-A01) and (PATS-A02). The letters and numbers attached to the names are used for possible future expansions.

3.5.1 The Source of the Selected Text

Most of the text used to prepare both datasets PATS-A01 and PATS-02 for Arabic text recognition was extracted from the books of *Saheh Al-Bukhari* [*132*] and *Saheh Muslem* [*133*]. The text of the books represents samples of standard classical Arabic. The extracted data were chosen to fairly represent Standard Arabic text of alphabets.

3.5.2 Dataset Descriptions

The first data set (PATS-A01) consists of 2766 text line images. The text of 2751 line images of this set was selected from the above books. The text of the remaining 15 line images are added from our minimal Arabic script which will be described in Section 3.8. The second data set (PATS-A02) is a subset of the first one. It consists of only 318 carefully chosen line images.

For each dataset, eight Microsoft Word document files with the same text were created, each with one of the eight used fonts. The used fonts were: Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic. Table 3-12 shows a sample for each font. The size of the used fonts was chosen to be 18. Each file was printed on paper sheets. The paper sheets were scanned into images representing the printed pages. Each file is also saved in “pdf” format and converted into “tif” images where each “tif” image represented a single line of text. At the end we have 2766 images representing 2766 text lines for each font of the eight fonts. We have also the scanned pages of the printed formatted text for each font. The ground truth information is represented as a Unicode text file.

For each image file representing a text line, the image was converted to binary format (i.e. white text on black background). Moreover, each image in the 'tif' file has been mirrored as shown in Figure 3.1. The mirroring is used for compatibility with left-to-right languages as most of the programming languages and tools assume left-to-right layouts.

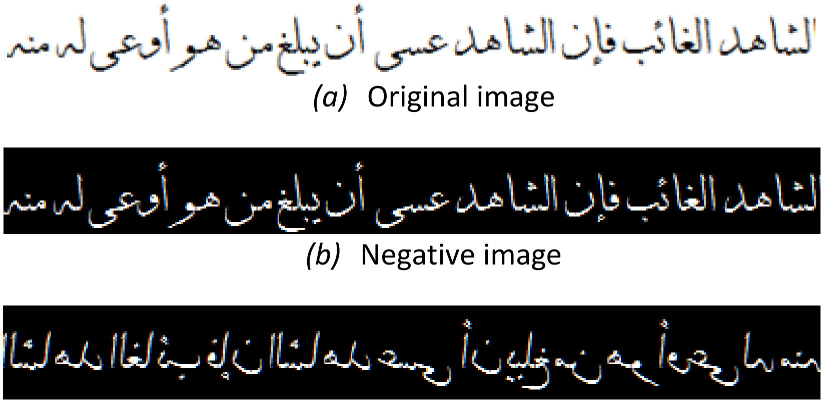
Out of the 2766 line images, 15 line images were added to assure the inclusion of a sufficient number of all shapes of Arabic letters. These lines consist of 5 copies of the minimal Arabic script that we have developed for preparing databases and benchmarks for Arabic text recognition research (See Al-Muhtaseb et al. [*134*] and [*135*]). The minimal Arabic script will be discussed in Section 3.8.

Table 3-12: Samples of all fonts used.

Font Name	Sample
Arial	حسن الخلق من الإيمان
Tahoma	حسن الخلق من الإيمان
Akhbar	حسن الخلق من الإيمان
Thuluth	حسن الخلق من الإيمان
Naskh	حسن الخلق من الإيمان
Simplified Arabic	حسن الخلق من الإيمان
Traditional Arabic	حسن الخلق من الإيمان
Andalus	حسن الخلق من الإيمان

3.6 Dataset Statistics

Dataset PATS-A01 consists of 46062 words totalling 224109 characters including spaces. The average word length of the text is 3.93 characters. Words are separated by spaces. There are no two consecutive spaces in any line. The length of the smallest line is 43 characters. The longest line has 89 characters. Table 3-13 and Figure 3.2 show the frequencies of characters in this dataset. The frequency distribution differs from character to character depending on its natural distribution in classic standard Arabic, although this varies from domain to domain. Some characters are naturally used more than other characters. The letters Alef (ا) and Lam (ل) frequently have high frequencies in any representative text. Each of these two letters might represent 10% of the text.


لشاهد الغائب فإن الشاهد عسى أن يبلغ من هو أوعى له منه

(a) Original image

(b) Negative image

(c) Mirrored image

Figure 3.1: An example of a line image.

Table 3-14 shows the frequencies of each shape of the Arabic letters in PATS-A01 for one of the used fonts. It is worth pointing out that in the letters Alef (ا) and Lam (ل) the sum of all shapes of each letter will not add to the total number in Table 3-13 as part of these two letters are also distributed on the *LamAlef* shapes (لا). A similar thing should be noticed with different ligatures including the letter *Alef* with *Hamza*, depending on the used fonts and added ligatures.

The PATS-A02 dataset is a subset of the PATS-A01 dataset. The aim of this dataset is to have a smaller data that still carry the characteristics of the Standard classical Arabic. A smaller dataset could be very useful when multi-fonts are considered. The PATS-A02 dataset consists of 5771 words totalling 27486 characters including spaces. The average word length of the text is 3.82 characters. It has only 318 line images. 15 of them represent 5 copies of the minimal Arabic script. Table 3-15 and Figure 3.3 show the character distribution of the dataset PATS-A02. The frequencies of each shape of the Arabic letters in PATS-A02 are shown in Table 3-16.

Table 3-13: Character distribution of dataset PATS-A01.

Letter	Frequency	Percentage	Letter	Frequency	Percentage
	43296	19.32	س	5279	2.36
ء	760	0.34	ش	1142	0.51
آ	219	0.1	ص	3140	1.4
أ	5505	2.46	ض	979	0.44
ؤ	209	0.09	ط	844	0.38
إ	1795	0.8	ظ	282	0.13
ئ	475	0.21	ع	7322	3.27
ا	21923	9.78	غ	658	0.29
ب	7586	3.38	ف	5562	2.48
ة	2351	1.05	ق	4937	2.2
ت	5082	2.27	ك	3522	1.57
ث	1573	0.7	ل	25342	11.31
ج	2152	0.96	م	10882	4.86
ح	2667	1.19	ن	11192	4.99
خ	1234	0.55	ه	9051	4.04
د	3674	1.64	و	9072	4.05
ذ	1600	0.71	ى	3191	1.42
ر	6988	3.12	ي	11976	5.34
ز	647	0.29	Total	224109	100

The distribution is still a fair representation of standard Arabic statistics, where characters with low frequencies still have enough samples for training. The lowest frequency in this dataset is 22, which is for the letter *Hamza over Waw* **(ؤ).** This letter naturally has low appearance in the language. It has two basic shapes: isolated and connected from right. 22 instances of this letter are enough for the training process.

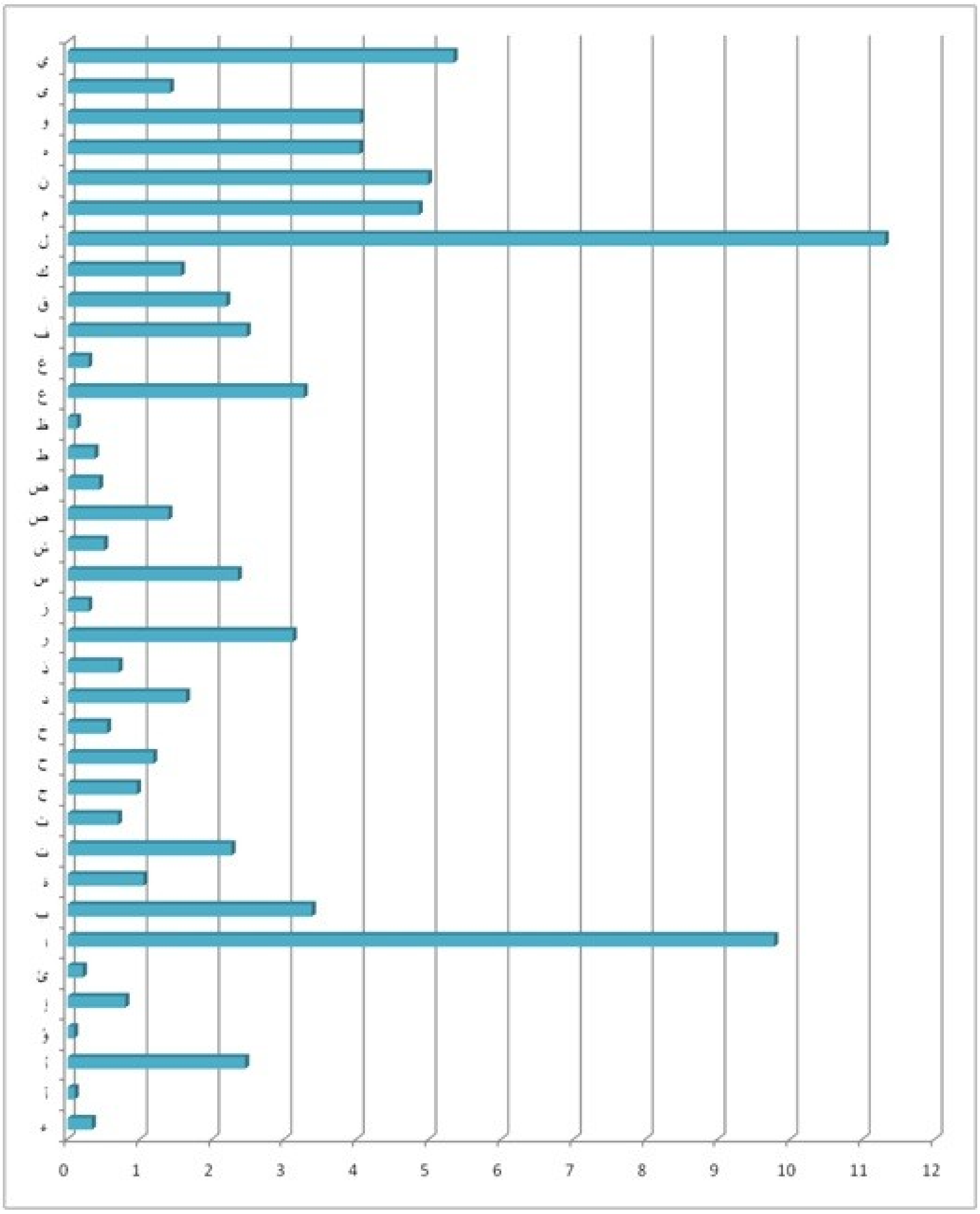


Figure 3.2: Frequency distribution graph for dataset PATS-A01

Table 3-14: Shape distribution of dataset PATS-A01.

Letter	Shape	Freq.	Letter	Shape	Freq.	Letter	Shape	Freq.	Letter	Shape	Freq.	Letter	Shape	Freq.
ء	ء	760	ث	ـث	120	س	سـ	3022	غ	غ	36	ن	ن	1941
آ	آ	153	ث	ـثـ	393	ش	ش	15	غ	ـغ	34	ن	ـن	3563
آ	ـآ	12	ث	ثـ	984	ش	ـش	50	غ	ـغـ	300	ن	ـنـ	3652
أ	أ	4329	ج	ج	167	ش	ـشـ	591	غ	غـ	288	ن	نـ	2036
أ	ـأ	825	ج	ـج	47	ش	شـ	486	ف	ف	138	ه	ه	606
ؤ	ؤ	49	ج	ـجـ	868	ص	ص	14	ف	ـف	184	ه	ـه	3382
ؤ	ـؤ	160	ج	جـ	1070	ص	ـص	36	ف	ـفـ	738	ه	ـهـ	1626
إ	إ	1442	ح	ح	42	ص	ـصـ	1252	ف	فـ	4502	ه	هـ	1185
إ	ـإ	218	ح	ـح	210	ص	صـ	1838	ق	ق	98	و	و	5464
ئ	ئ	15	ح	ـحـ	952	ض	ض	73	ق	ـق	161	و	ـو	3608
ئ	ـئ	22	ح	حـ	1463	ض	ـض	135	ق	ـقـ	1821	لآ	لآ	49
ئ	ـئـ	152	خ	خ	7	ض	ـضـ	323	ق	قـ	2857	لآ	ـلآ	5
ئ	ئـ	286	خ	ـخ	23	ض	ضـ	448	ك	ك	112	لأ	لأ	334
ا	ا	9911	خ	ـخـ	441	ط	ط	21	ك	ـك	643	لأ	ـلأ	17
ا	ـا	10497	خ	خـ	763	ط	ـط	70	ك	ـكـ	1091	لإ	لإ	128
ب	ب	418	د	د	773	ط	ـطـ	540	ك	كـ	1676	لإ	ـلإ	7
ب	ـب	390	د	ـد	2901	ط	طـ	213	ل	ل	2926	لا	لا	680
ب	ـبـ	2777	ذ	ذ	897	ظ	ظ	7	ل	ـل	1399	لا	ـلا	835
ب	بـ	4001	ذ	ـذ	703	ظ	ـظ	21	ل	ـلـ	6871	ى	ى	178
ة	ة	542	ر	ر	2424	ظ	ـظـ	193	ل	لـ	7587	ى	ـى	3013
ة	ـة	1809	ر	ـر	4564	ظ	ظـ	61	م	م	704	ي	ي	400
ت	ت	314	ز	ز	222	ع	ع	106	م	ـم	3398	ي	ـي	3142
ت	ـت	1346	ز	ـز	425	ع	ـع	559	م	ـمـ	3306	ي	ـيـ	4904
ت	ـتـ	2043	س	س	383	ع	ـعـ	2216	م	مـ	3474	ي	يـ	3530
ت	تـ	1379	س	ـس	335	ع	عـ	4441	ﷲ	ﷲ	2252			
ث	ث	76	س	ـسـ	1539	**Blank**	**Blank**	43296						

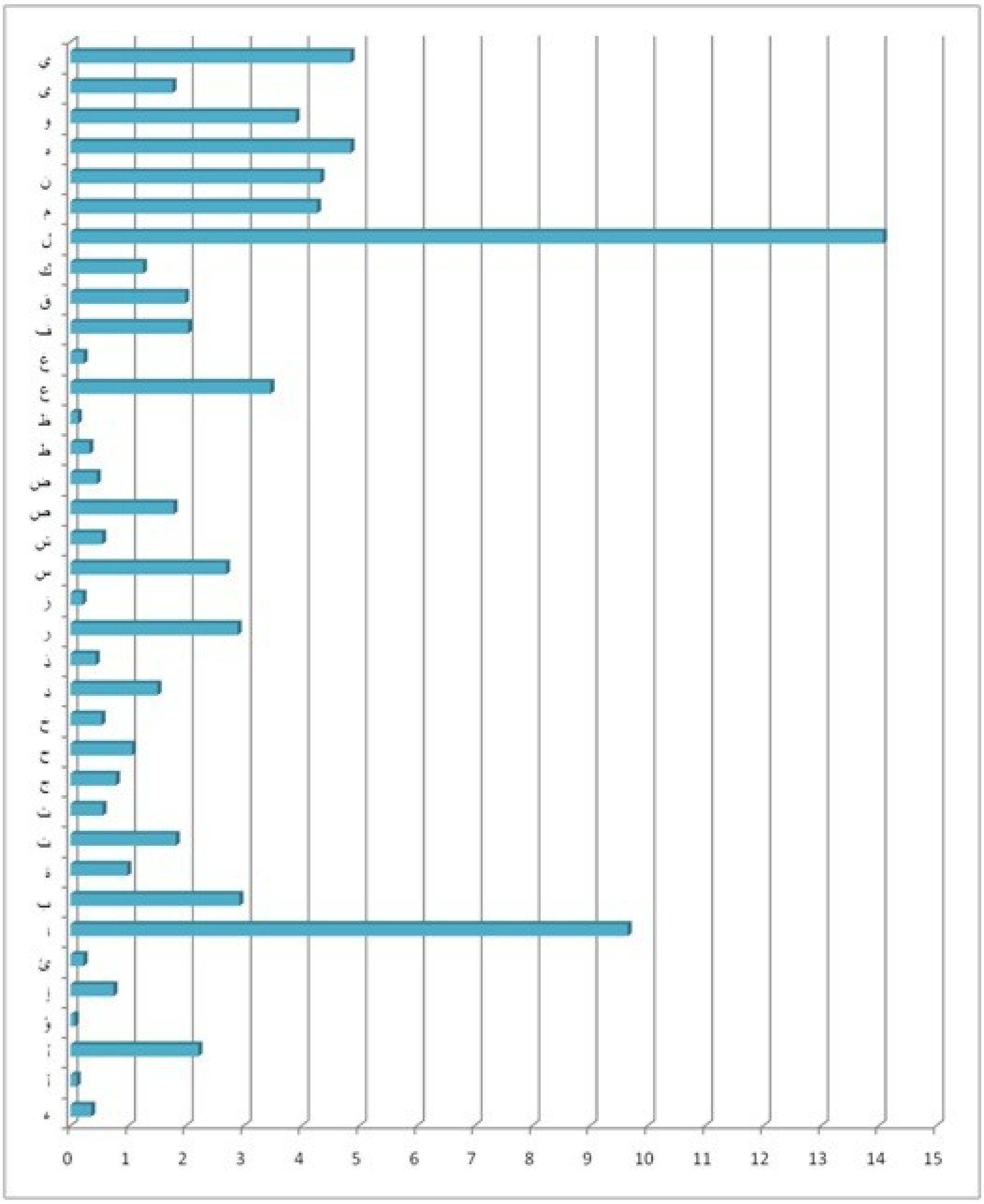


Figure 3.3: Frequency distribution graph for dataset PATS-A02.

Table 3-15: Character distribution of dataset PATS-A02.

Letter	Frequency	Percentage	Letter	Frequency	Percentage
	5453	19.84	س	741	2.7
ء	100	0.36	ش	150	0.55
آ	30	0.11	ص	492	1.79
أ	610	2.22	ض	126	0.46
ؤ	22	0.08	ط	91	0.33
إ	206	0.75	ظ	37	0.13
ئ	62	0.23	ع	951	3.46
ا	2656	9.66	غ	63	0.23
ب	805	2.93	ف	562	2.04
ة	273	0.99	ق	547	1.99
ة	504	1.83	ك	345	1.26
ث	153	0.56	ل	3870	14.08
ج	216	0.79	م	1174	4.27
ح	293	1.07	ن	1191	4.33
خ	149	0.54	ه	1332	4.85
د	414	1.51	و	1072	3.9
ذ	122	0.44	ى	486	1.77
ر	798	2.9	ي	1333	4.85
ز	57	0.21	Total	27486	100

3.7 Data Labelling

Researchers working on Arabic text recognition are accustomed to using Arabic letters as the basic unit of classification. In this research work we are using the shape of the letter as the basic unit of classification. In the former method, all different shapes of an Arabic letter is considered as one class. In our method, an Arabic letter with four basic shapes is given four different classes: a class for each shape. In the recognition experiments we are using the ground truth information to represent each letter shape differently. For example, the letter *Baa* (ب) has four basic shapes (See Figure 3.4) with a unique Unicode representation (U0633). In our own labelling, we gave each basic shape for every letter a different label. After recognition, we map the recognized characters to their unique Unicode representations. Software tools were developed for labelling, coding, and encoding.

3.8 Minimal Arabic Script

The novel idea, which is being introduced here, is to use a script that consists of a minimum number of letters (using meaningful Arabic words) covering all possible shapes. Although the main objective is to cover all shapes of Arabic letters, finding meaningful words containing these shapes is a second objective. The minimal Arabic script may be used for preparing databases and benchmarks for Arabic optical character recognition. This script will be very useful when soliciting volunteers to write some text for a handwritten database. It is much easier to ask a person to take part in the formation of a handwritten database when he/she has to write three lines only (not several pages as in [*65*]). The characteristics of the Arabic minimal script we are proposing are:

- covering all basic shapes of Arabic letters,
- using as minimal text as possible, and
- using meaningful words.

Table 3-16: Shape distribution of dataset PATS-A02.

Letter	Shape	Frq.	Letter	Shape	Frq.	Letter	Shape	Frq.	Letter	Shape	Frq.	Letter	Shape	Frq.
ء	ء	100	ث	ـث	12	س	ـسـ	135	غ	غ	6	ن	ن	228
آ	آ	20	ث	ـثـ	29	س	سـ	523	غ	ـغ	1	ن	ـن	385
أ	أ	478	ث	ثـ	99	ش	ـش	8	غ	ـغـ	27	ن	ـنـ	362
أ	ـأ	81	ج	ج	19	ش	ـشـ	77	غ	غـ	29	ن	نـ	216
ؤ	ؤ	5	ج	ـج	9	ش	شـ	65	ف	ف	19	ه	ه	65
ؤ	ـؤ	17	ج	ـجـ	83	ص	ص	8	ف	ـف	20	ه	ـه	997
إ	إ	164	ج	جـ	105	ص	ـص	9	ف	ـفـ	82	ه	ـهـ	155
إ	ـإ	21	ح	ح	9	ص	ـصـ	143	ف	فـ	441	ه	هـ	115
ئ	ئ	1	ح	ـح	20	ص	صـ	332	ق	ق	5	و	و	636
ئ	ـئ	8	ح	ـحـ	98	ض	ض	15	ق	ـق	21	و	ـو	436
ئ	ـئـ	15	ح	حـ	166	ض	ـض	13	ق	ـقـ	213	لآ	لآ	5
ئ	ئـ	38	خ	خ	5	ض	ـضـ	35	ق	قـ	308	لآ	ـلآ	5
ا	ا	1300	خ	ـخ	5	ض	ضـ	63	ك	ك	14	لأ	لأ	43
ا	ـا	1121	خ	ـخـ	59	ط	ط	3	ك	ـك	65	لأ	ـلأ	8
ب	ب	33	خ	خـ	80	ط	ـط	10	ك	ـكـ	96	لإ	لإ	21
ب	ـب	38	د	د	80	ط	ـطـ	48	ك	كـ	170	لإ	لإ	123
ب	ـبـ	321	د	ـد	334	ط	طـ	30	ل	ل	428	لا	ـلا	112
ب	بـ	413	ذ	ذ	66	ظ	ظ	5	ل	ـل	146	ى	ى	24
ة	ة	123	ذ	ـذ	56	ظ	ـظ	6	ل	ـلـ	1627	ى	ـى	462
ة	ـة	150	ر	ر	361	ظ	ـظـ	20	ل	لـ	1352	ي	ي	46
ت	ت	37	ر	ـر	437	ظ	ظـ	6	م	م	92	ي	ـي	359
ت	ـت	143	ز	ز	14	ع	ع	13	م	ـم	428	ي	ـيـ	560
ت	ـتـ	173	ز	ـز	43	ع	ـع	46	م	ـمـ	314	ي	يـ	368
ت	تـ	151	س	س	38	ع	ـعـ	225	م	مـ	340	**Blank**	**Blank**	5453
ث	ث	13	س	ـس	45	ع	عـ	667						

Moreover, the images of the words of the minimal Arabic script have been used to thoroughly study the characteristics of the shapes of Arabic letters in order to introduce a new discriminating feature extraction scheme (see Section 4.2).

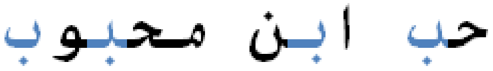
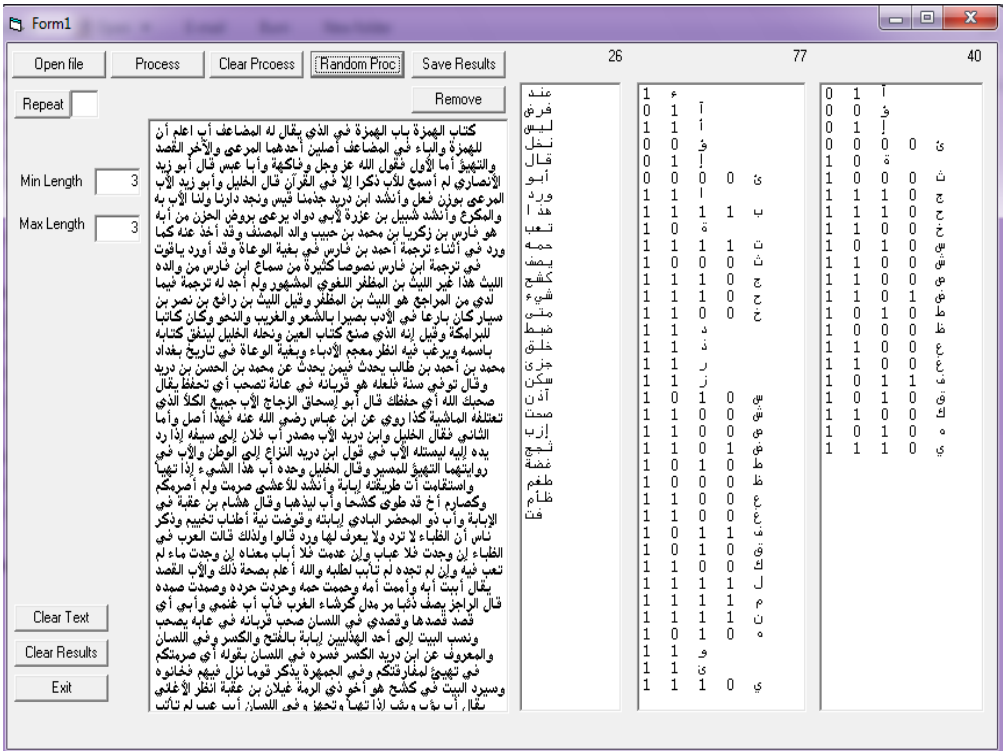


Figure 3.4: The four different basic shapes of the letter *Baa* (ب).

Several utility programs, implementing different algorithms to address this issue, were developed to search huge corpora of Arabic script to find a set of minimum number of meaningful words that cover all Arabic alphabet-shapes. Figure 3.5 shows the user interface of one of these utilities. The utility software was designed to get words from any chosen file with Unicode format text. It allows the user to experiment with different options. For each process it displays the status of covered shapes of different Arabic letters. The utility along with its source code are provided in the enclosed CD-ROM (See Appendix A).

3.8.1 Used Corpora for the Minimal Arabic Script

The used corpora for our analysis consists of Arabic text of two Arabic lexicons [*136*] [*137*], two *HADITH* books [*132*], [*133*], and a lexicon containing the meaning of Quran tokens in Arabic [*138*]. The electronic versions of such books and other old Arabic classical books can be found in different websites including [*139*].



Min Length 3
Max Length 3

كتاب الهمزة باب الهمزة في الذي يقال له المضاعف أب اعلم أن
للهمزة والباء في المضاعف أصلين أحدهما المرعى والآخر القصد
والتهيؤ أما الأول فقول الله عز وجل وفاكهة وأبا عبس قال أبو زيد
الأنصاري لم أسمع للأب ذكرا إلا في القرآن قال الخليل وأبو زيد الأب
المرعى بوزن فعل وأنشد ابن دريد جذمنا قيس ونجد دارنا ولنا الأب به
والمكرع وأنشد شبيل بن عزرة لأبي دواد يرعى بروض الحزن من أبه
هو فارس بن زكريا بن محمد بن حبيب والد المصنف وقد أخذ عنه كما
ورد في أثناء ترجمة أحمد بن فارس في بغية الوعاة وقد أورد ياقوت
في ترجمة ابن فارس نصوصا كثيرة من سماع ابن فارس من والده
الليث هذا غير الليث بن المظفر اللغوي المشهور ولم أجد له ترجمة فيما
لدي من المراجع هو الليث بن المظفر وقيل الليث بن رافع بن نصر بن
سيار كان بارعا في الأدب بصيرا بالشعر والغريب والنحو وكان كاتبا
للبرامكة وقيل إنه الذي صنع كتاب العين ونحله الخليل لينفق كتابه
باسمه ويرغب فيه انظر معجم الأدباء وبغية الوعاة في تاريخ بغداد
محمد بن أحمد بن طالب يحدث فيمن يحدث عن محمد بن الحسن بن دريد
وقال توفي سنة فلعله هو قريانه في عانة تصحب أي تحفظ يقال
صحبك الله أي حفظك قال أبو إسحاق الزجاج الأب جميع الكلأ الذي
تعتلفه الماشية كذا روي عن ابن عباس رضي الله عنه فهذا أصل وأما
الثاني فقال الخليل وابن دريد الأب مصدر أب فلان إلى سيفه إذا رد
يده إليه ليستله الأب في قول ابن دريد النزاع إلى الوطن والأب في
روايتهما التهيؤ للمسير وقال الخليل وحده أب هذا الشيء إذا تهيأ
واستقامت أت طريقته إيابة وأنشد للأعشى صرمت ولم أصرمكم
وكصارم أخ قد طوى كشحا وأب ليذهبا وقال هشام بن عقبة في
الإبابة وأب ذو المحضر البادي إبابته وقوضت نية أطناب تخييم وذكر
ناس أن الظباء لا ترد ولا يعرف لها ورد قالوا ولذلك قالت العرب في
الظباء إن وجدت فلا عباب وإن عدمت فلا أباب معناه إن وجدت ماء لم
تعب فيه وإن لم تجده لم تأبب لطلبه والله أعلم بصحة ذلك والأب القصد
يقال أببت أبه وأممت أمه وحممت حمه وحردت حرده وصمدت صمده
قال الراجز يصف ذئبا مر مدل كرشاء الغرب فأب أب غنمي وأبي أي
قصد قصدها وقصدي في اللسان صحب قريانه في عابه يصحب
ونسب البيت إلى أحد الهذليين إبابة بالفتح والكسر وفي اللسان
والمعروف عن ابن دريد الكسر فسره في اللسان بقوله أي صرمتكم
في تهيئ لمفارقتكم وفي الجمهرة يذكر قوما نزل فيهم فخانوه
وسيرد البيت في كشح هو أخو ذي الرمة غيلان بن عقبة انظر الأغاني
يقال أب يؤب ويئب إذا تهيأ وتجهز وفي اللسان أبب عب لم تأتب

```
26
عند
فرض
ليس
نخل
قال
أبو
ورد
هذ ا
تعب
حمه
يصف
كشح
شيء
متى
ضبط
خلق
جزى
سكن
آذن
صحت
إزب
ثجج
غضة
طغم
ظأم
فت
```

```
77
1   ء
0   1   آ
1   1   أ
0   0   ؤ
0   1   إ
0   0   0   0   ئ
1   1   ا
1   1   1   1   ب
1   0   ة
1   1   1   1   ت
1   0   0   0   ث
1   1   1   0   ج
1   1   1   0   ح
1   1   0   0   خ
1   1   د
1   1   ذ
1   1   ر
1   1   ز
1   0   1   0   س
1   1   0   0   ش
1   1   0   0   ص
1   1   0   1   ض
1   0   1   0   ط
1   0   0   0   ظ
1   1   0   0   ع
1   1   0   0   غ
1   0   1   1   ف
1   0   1   0   ق
1   1   0   0   ك
1   1   1   1   ل
1   1   1   1   م
1   1   1   1   ن
1   0   1   0   ه
1   1   و
1   1   ى
1   1   1   0   ي
```

```
40
0   1   آ
0   0   ؤ
0   1   إ
0   0   0   0   ئ
1   0   ة
1   0   0   0   ث
1   1   1   0   ج
1   1   1   0   ح
1   1   0   0   خ
1   0   1   0   س
1   1   0   0   ش
1   1   0   0   ص
1   1   0   1   ض
1   0   1   0   ط
1   0   0   0   ظ
1   1   0   0   ع
1   1   0   0   غ
1   0   1   1   ف
1   0   1   0   ق
1   1   0   0   ك
1   0   1   0   ه
1   1   1   0   ي
```

Figure 3.5: The user interface of the software tool to semi automate finding a minimal Arabic script.

3.8.2 Determining a Minimal Arabic Script

It was clear from the literature review that there are no adequate Arabic text databases freely available for use in the research of Arabic typewritten text. The produced minimal text and the work presented in [*140*] (which presented the probabilities of the occurrence of the Arabic alphabets in different positions of Arabic words), is an efficient solution to the above problem, which is belived to be a new contribution to the field.

As illustrated earlier, Arabic text is saved using a unique code for each character irrespective of its position and shape. When the statistics of a certain Arabic character shapes are required, a procedure is used to identify these shapes. An algorithm was implemented to decode and tag the letters with their positions code in the word according to the context of the word (initial. medial, terminal, or isolated). The classes used in the algorithm are those that are presented earlier and summarized in Table 3-17.

Table 3-17: Classes of Arabic alphabets depending on number of possible basic shapes.

Class	# of possible shapes	Alphabets
1	1	ء
2	2	آ أ ؤ إ ا ة د ذ ر ز و لآ لأ لإ لا ى
3	4	ئ ب ت ث ج ح خ س ش ص ض ط ظ ع غ ف ق ك ل م ن ه ي

Figure 3.6 shows the pseudo-code of an algorithm (*processWord*) to process words extracted from Arabic corpora to generate the minimal text. These are the main steps of the *processWord* algorithm.

1. Initially the word is validated to see if it is already in the minimal text, if this is true then the word is not processed and the search for more words continues.
2. The word is decoded to give the proper letter shapes of the word using the implemented contextual analysis algorithm.
3. The word is validated for multiple occurrences of a letter, if it has multiple occurrences of a letter then the word is not processed and the search proceeds for a new word, as repetition is not allowed.
4. Each letter of the word is checked with the letters' table (holding the different shapes of Arabic alphabet). If any letter in the word is already flagged in the letters' table then the word is ignored and the search proceeds for new words.
5. If a word passes the previous validations then
    a. The word is added to the minimal text, and
    b. The shapes of the all-shapes table corresponding to the letters of the word are flagged.

```
Definition of used variable/parameters:
aword: Arabic word
wordShapes: List of aword letters with specific shapes
element: a letter from wordShapes
minTextTable: A table holding minimum text
alphabetTable: Arabic alphabet table including extra column for
                                  flagging used letters.
characerTagged: A flag to indicate tagging of letters

  function processWord(aword)
  { //checks if the word is already in minTextTable
    if(aword is in minTextTable)
      exit;

      Decode the word into letter shapes and put them in
                               wordShapes
  // Each letter of the word is given letter and shape code by
  //the implemented contextual analysis algorithm.

    charTagged = 0;   // initialize charTagged flag to 0

    for(i=0;i< count(wordShapes);i++)(
      element = wordShapes[i]
         If(element is flagged in alphabetTable){
                      //Was this letter used?
            charTagged=1;
            break;
 }
}

 if(not(charTagged)){
      Add word to minTextTable;   //add aword to minimal text
     for(i=0;i< count(wordShapes);i++){
                      //tag aword letters in alphabetTable
  element = wordShapes[i]
          flag element in alphabetTable;
     }
 charTagged = 0; // clear tagging flag
}

}
```

Figure 3.6: Pseudo-code for processing Arabic words for the generation of minimal text to cover all Arabic letters in all positions in a word.

Several search criteria are conducted on the corpora to generate the minimal Arabic text using the *processWord* function. The function *processWord* starts by sequentially searching the corpora for targeted words. This process continues until the whole corpora are searched. Then the resulting letters' table is checked for un-flagged letters. It is clear that this process could not flag all shapes and hence the minimum text does not cover all shapes. In a second version of the function, different sequences of the data in the corpora are selected (i.e. the sequence of searching the corpora is changed several times). This does not result in acquiring a minimum text to cover all Arabic alphabet shapes. In a third version of the function, the words are randomly selected from the corpora. This showed better results, however, the minimal produced text does not include all the Arabic alphabet shapes. Another search algorithm is executed which starts by selecting words having letters of minimal frequencies of usage utilizing the estimated frequencies of Arabic alphabet in the Arabic script. Hence, less frequently used letters are given higher priority. This results in improvements to the minimal produced text. In all of these experiments there is a constraint of not using a letter shape more than one time.

By analyzing the different shapes of Arabic alphabets, it can be observed that there are 39 shapes of letters that might come at the end of a word in terminal form and 23 shapes of the letters that might come at the beginning of a word in initial form. Hence, there should be some repetitions of the letters' shapes that come at the beginning in order to include all the shapes of Arabic letters. The previous search algorithms were applied again, allowing the possibility of an initial letter at the beginning to have up to two occurrences. In addition to these constraints, we limited the total number of extra occurrences of these letters to 16. By using a corpora of around 20 Megabytes of text and using the programs we have developed, we could reach a nearly optimal script. An early minimal script has been identified as shown in Figure 3.7. The script then was optimized manually through several iterations until it reached its existing structure as shown in Figure 3.8. The manual optimization was used to include few shapes that were not included after the exhaustive automatic optimization.

Table 3-18 shows the statistics of letter shape distribution for the suggested minimal Arabic script. It is clear from the table that 16 initial letter shapes have two occurrences each to compensate for the extra shapes of Arabic letter shapes that come at the end of a word in terminal position. All other letter shapes are used only once. Hence, the presented text is the minimal possible text that covers all the basic shapes of Arabic alphabets. It is minimal in terms of the number of shapes used.

جعثق سؤق ذم ظأب رث مجس غضبى قمين شتف وس ضغط أي بحظل حذف نسكه
طخفة خصهم صئك إظ فت زؤى كنب دك آخ ال هوج ثآئي طء لطاع يقره عزة تشدح
لاغ لأص لآت لإض ش ئ مج حث جح صخ فإن يجئ نص قش عض لظ بلغ سع

Figure 3.7: An early minimal Arabic script.

It might be clear to the reader that the minimal script is not unique. Theoretically speaking, there is infinite number of different scripts. A main characteristic of all these minimal scripts is that they all should have only 141 Arabic letter basic shapes that cover all Arabic letter shapes (see Section 4.2).

عزة كأب جنة طفق ميس غضبى كثف ضغط أص فظل حذف نسكه خصتهم صئك إظ رؤى
شعب دك آخ ملأ سطوع يقرء تشدح لاغ للآت لإض هج حث جح صخ فإي يجئ نخص قش
عض تحظ بلغ سع ظمآن قن طائي ثلاث لأج لآه بؤس ذم ائت للإوز ق ط ش ل ئ

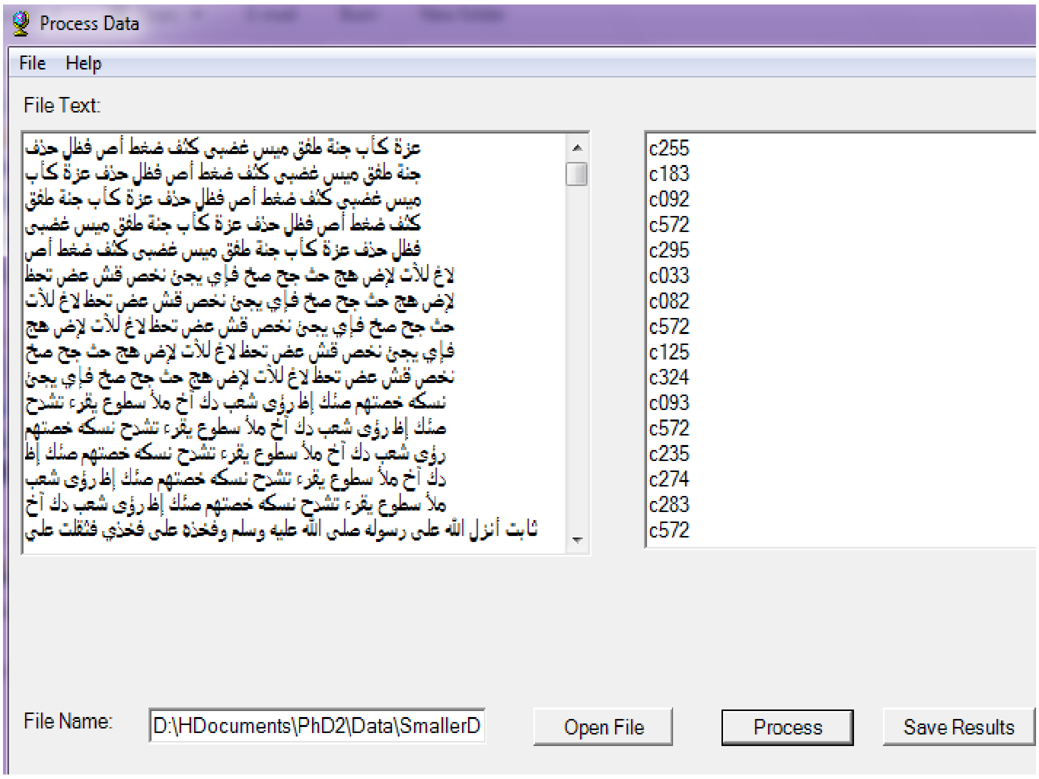
Figure 3.8: The minimal Arabic script.

Table 3-18: Minimal text usage of the different shapes of Arabic letters.

Letter	Standalone	Terminal	Initial	Medial
ء	1			
آ	1	1		
أ	1	1		
ؤ	1	1		
إ	1	1		
ئ	1	1	2	1
ا	1	1		
ب	1	1	2	1
ة	1	1		
ت	1	1	2	1
ث	1	1	1	1
ج	1	1	2	1
ح	1	1	2	1
خ	1	1	1	1
د	1	1		
ذ	1	1		
ر	1	1		
ز	1	1		
س	1	1	2	1
ش	1	1	1	1
ص	1	1	2	1
ض	1	1	1	1
ط	1	1	2	1
ظ	1	1	1	1
ع	1	1	2	1
غ	1	1	1	1
ف	1	1	2	1
ق	1	1	2	1
ك	1	1	2	1
ل	1	1	2	1
م	1	1	2	1
ن	1	1	2	1
ه	1	1		
و	1	1		
لآ	1	1		
لأ	1	1		
لإ	1	1		
لا	1	1		
ى	1	1	1	1
ي	1	1	2	1

3.9 Coding and Decoding

For compatibility issues, we have developed a software tool to convert the ground truth information of the text under experiment from Unicode format to a special-purpose format that will be used throughout our experiments. In the special-purpose format we coded each shape of every letter by a unique code. The developed software tool has the capability to decode back the recognized text from the special-purpose format to the Unicode format. The user interface of the tool is shown in Figure 3.9. The tool and its source code are provided in the enclosed CD-ROM (See Appendix A).



D:\HDocuments\PhD2\Data\SmallerD

عزة كأب جنة طفق ميس غضبى كنف ضغط أص فظل حذف
جنة طفق ميس غضبى كنف ضغط أص فظل حذف عزة كأب
ميس غضبى كنف ضغط أص فظل حذف عزة كأب جنة طفق
كنف ضغط أص فظل حذف عزة كأب جنة طفق ميس غضبى
فظل حذف عزة كأب جنة طفق ميس غضبى كنف ضغط أص
لاغ للأت لإض هج حث جح صخ فإي يجئ نخص قش عض تحظ
لإض هج حث جح صخ فإي يجئ نخص قش عض تحظ لاغ للأت
حث جح صخ فإي يجئ نخص قش عض تحظ لاغ للأت لإض هج
فإي يجئ نخص قش عض تحظ لاغ للأت لإض هج حث جح صخ
نخص قش عض تحظ لاغ للأت لإض هج حث جح صخ فإي يجئ
نسكه خصتهم صئك إظ رؤى شعب دك آخ ملأ سطوع يقرء تشدح
صئك إظ رؤى شعب دك آخ ملأ سطوع يقرء تشدح نسكه خصتهم
رؤى شعب دك آخ ملأ سطوع يقرء تشدح نسكه خصتهم صئك إظ
دك آخ ملأ سطوع يقرء تشدح نسكه خصتهم صئك إظ رؤى شعب
ملأ سطوع يقرء تشدح نسكه خصتهم صئك إظ رؤى شعب دك آخ
ثابت أنزل الله على رسوله صلى الله عليه وسلم وفخذه على فخذي فثقلت علي

```
c255
c183
c092
c572
c295
c033
c082
c572
c125
c324
c093
c572
c235
c274
c283
c572
```

Figure 3.9: The user interface for the coding/decoding tool.

3.10 Synthesized Data versus Scanned Data

The images of the text lines were prepared using two different methods. In the first method, computer programs were used to print different font files as images. In the second method, all text files were printed on paper with different fonts, and then a scanner was used to scan the printed pages and save them as images. Scanning printed pages was deliberately carried out to provide a real data. Both datasets were used in training and testing to assure the reality of the process.

3.11 Current Status of the Datasets

A website is being established to make the datasets available for research community. It can be reached online [*129*] through the link http://faculty.kfupm.edu.sa/ics/muhtaseb/ArabicOCR. We hope that this web site will be expanded in the future with more public Arabic datasets. Initially, the site will contain the two datasets PATS-A01 and PATS-A02 with their ground truth values for all the used fonts. It will contain the synthesized images as well as the scanned images. It will also include the results which we have got for each dataset for each font. The list of different training and testing sets used in our experiments will also be included to ensure possible accurate comparisons.

3.12 Summary and Conclusion

This chapter introduced the statistical analysis of two standard classical Arabic text books and presented the prepared printed Arabic datasets. The two books have 4,405,318 characters representing 1,095,274 words. The count of unique words is 50,367. The statistics were carried out mainly on the frequencies of different shapes of Arabic alphabets and written Arabic syllables.

These statistics can help in preparing suitable data that can fairly and naturally represent Arabic. The statistics could also be used for adding more accuracy while doing classifications in an Arabic OCR system by including a bigram language model. It can also be used in a post-processing phase following the classification phase to correct possible mistakes.

The detailed statistics are provided in the enclosed CD-ROM (See Appendix A).

Since there are no adequate benchmarks datasets for research on printed Arabic OCR, we have decided to tackle this problem by creating our own. We have introduced two datasets namely PATS-A01 and PATS-A02. The first dataset has 2766 line images representing 65062 words. The second dataset represents 5771 words making 318 line images. Each set of the two datasets contains enough samples of basic shapes of Arabic alphabets.

In each dataset, 5 copies of the developed minimal Arabic script were added to ensure the coverage of all basic shapes of the Arabic alphabets. The developed minimal Arabic script consists of a few Arabic words that contain all the basic shapes of all Arabic alphabets. The script could be also used to build an Arabic handwritten database as a benchmark. The script consists of only three lines. This encourages many volunteers to participate with their handwritings in the creation of handwritten benchmark databases.

The ground truth information of each line image is also available and considerable efforts were made to ensure 100% correctness. Such information represents the actual Arabic text of the line image.

Both datasets are available freely for researchers in both synthesized and scanned versions. Copies of the datasets along with their ground truth information and other related material are provided in the enclosed CD-ROM (See Appendix A).

Chapter 4. Feature Extraction

4.1 Introduction

The feature extraction phase has a crucial effect on the recognition rate of any OCR system [*141*]. Feature extraction is used to underline the distinctive properties of an object under consideration. The irrelevant data should be filtered in this stage.

This chapter introduces the feature extraction techniques that have been used in this research work to automatically recognize printed Arabic text. Section 4.2 highlights the individuality of Arabic alphabets as their discriminating properties will be extracted. Section 4.3 describes the general template of the proposed feature extraction scheme. Three applied cases of the proposed scheme are described in detail in sections 4.6, 4.5, and 4.4. The conclusion and summary are presented in Section 4.7.

4.2 Discriminating Characteristics of Arabic Letters

In any OCR system, a feature extraction phase should provide minimal representation for each character to capture its distinctive properties, or what is sometimes called the individualities of the characters. Figure 4.1 shows part of the images of Arabic characters. Those images along with the images of the remaining characters were used to thoroughly study the individualities of Arabic characters. All these images are provided in the enclosed CD-ROM (See Appendix A). Moreover, we have used the images of the minimal Arabic script that we have introduced in Chapter 3 to analyze the characteristics that discriminate Arabic letters from each other.

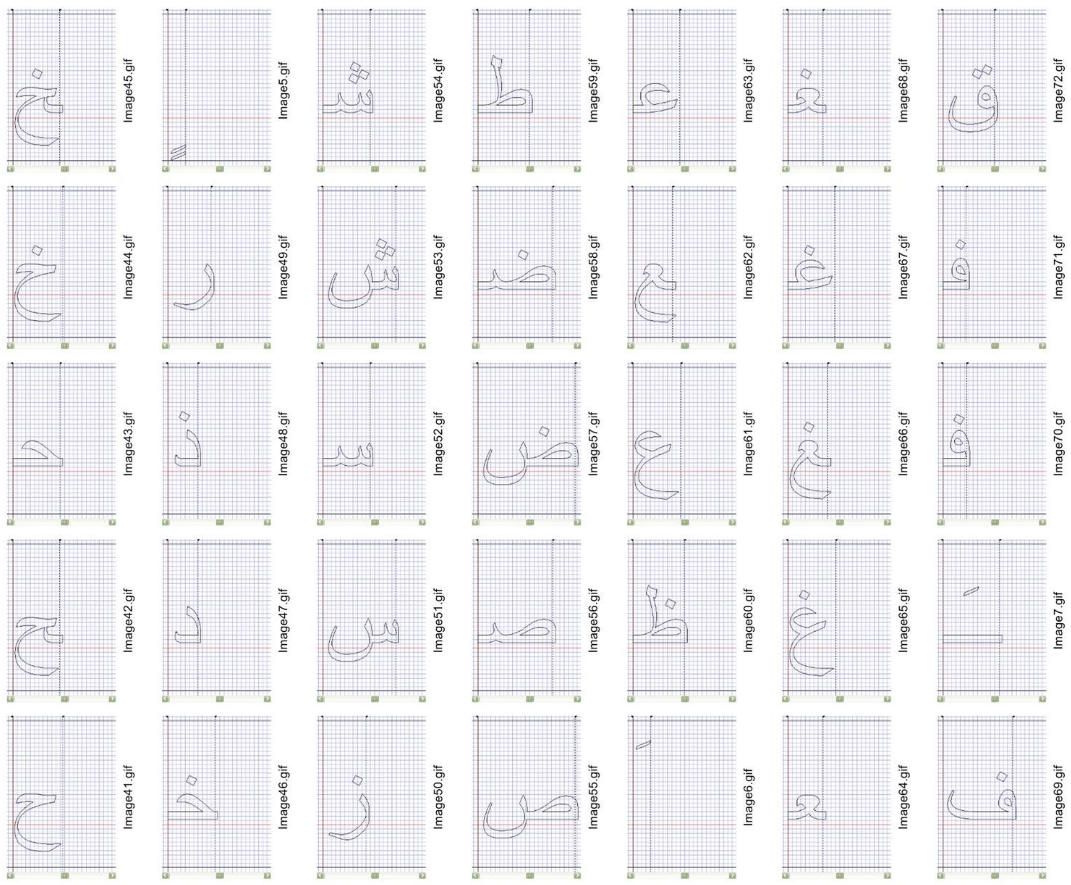


Figure 4.1: Part of Arabic characters.

Figure 4.2 shows several word images of the minimal Arabic script that were developed for this study. By studying the physical layout of Arabic alphabets, we notice that Arabic characters have different widths and different heights. All the letters in the Arabic alphabets have major parts of their shapes located above the baseline (see Section 1.2). The majority of the shapes of the letters don’t occupy more than one fourth of the height of the character above the baseline. Few shapes expand below the baseline. Also few other shapes expand above the central location of the character. Most shapes that expand below the baseline don’t expand above the central location of the characters. Very few shapes do expand above the middle of the size of the characters as well as below the baseline. These noticeable characteristics are simple guidelines to propose feature extraction schemes that highlight the individualities of Arabic alphabets. Some selective Arabic characters representing letter individualities are shown in Figure 4.3.

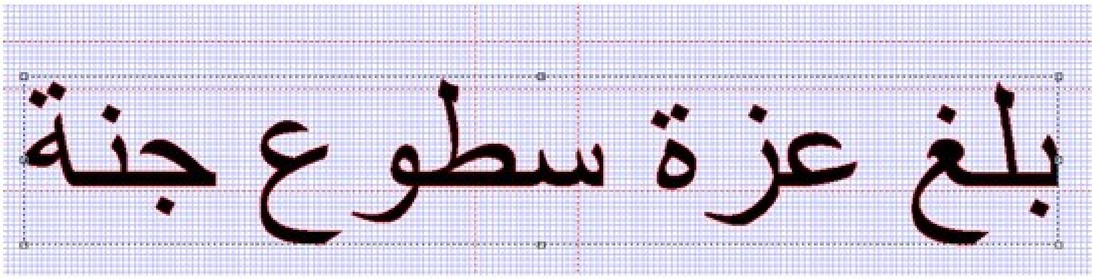


Figure 4.2: Image of some words of the minimal Arabic script.

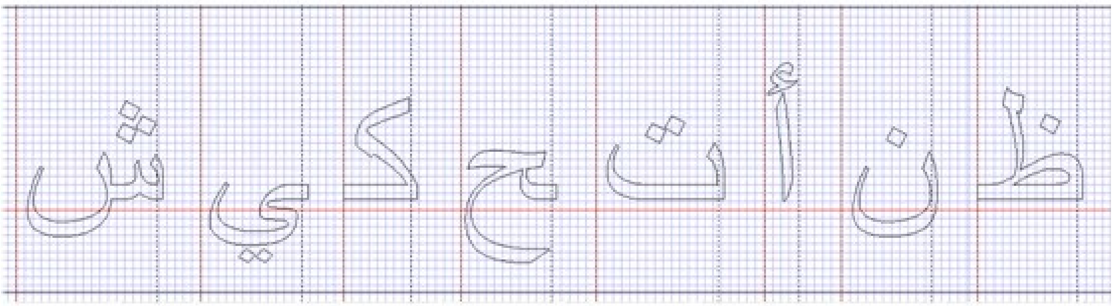


Figure 4.3: Selective Arabic characters representing letter individualities

4.3 The Proposed Feature Extraction Scheme

The proposed extraction scheme works on binary images and depends on two windows: W_H and W_V. W_H is a horizontally sliding window and W_V is a vertically moving variant window inside W_H. The width of W_H could be p pixels, where p is an integer number that is determined empirically. The height of this window is equal to h pixels, where h represents the height in pixels of the image under consideration. W_H slides horizontally from the beginning of the image till the end with q pixels overlapping, where q is an integer number less than the width of the window p. The vertically moving variant window W_V has a width of p pixels, the same width as W_H. The height of W_V is also found empirically. However, the basic proposed height of W_V is $k = h/n$, where h is the height of W_H in pixels and n represents the number of horizontal areas that will be used to define character individualities. For this case W_V slides vertically with no overlap for k times from the top of W_H till the bottom of the window. For each sliding iteration the pixels of the binary image with "1" values are counted and considered as one feature. So for the basic case of the proposed feature extraction scheme there should be at least k features per W_H slice. Figure 4.4 illustrates the basic definitions used above. The common scaling problem that might arise in such schemes is managed by image normalization.

The proposed feature extraction scheme could be used for other related applications such as handwriting recognition and other languages recognition. Some customization might be needed for this purpose. An example of such customization is to add more simple features. Suggested possible features to add could be the number of pixels with "1" values in each two consecutive W_V windows, three consecutive W_V windows, four consecutive W_V windows, and/or k consecutive W_V windows. Other possible features to be added could be the number of pixels with "1" values in each two $W_V(i)$ and $W_V(k\text{-}i\text{+}1)$ windows, where i starts from 1 till $k/2$. That is the first windows with last windows, the second windows with second window from last and so on.

Enough experimental cases were tested depending on the study of Arabic characters individualities. As a result of the experimental testing, several cases were proven to be good representations to be used in training and classifications as they produced better recognition. The next three sections show three working cases of the implementation of this feature extraction scheme.

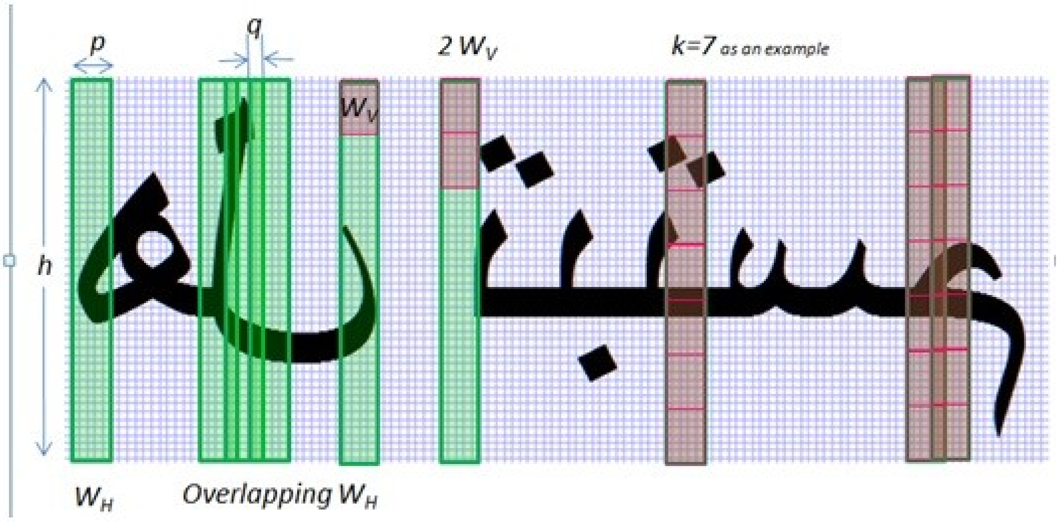
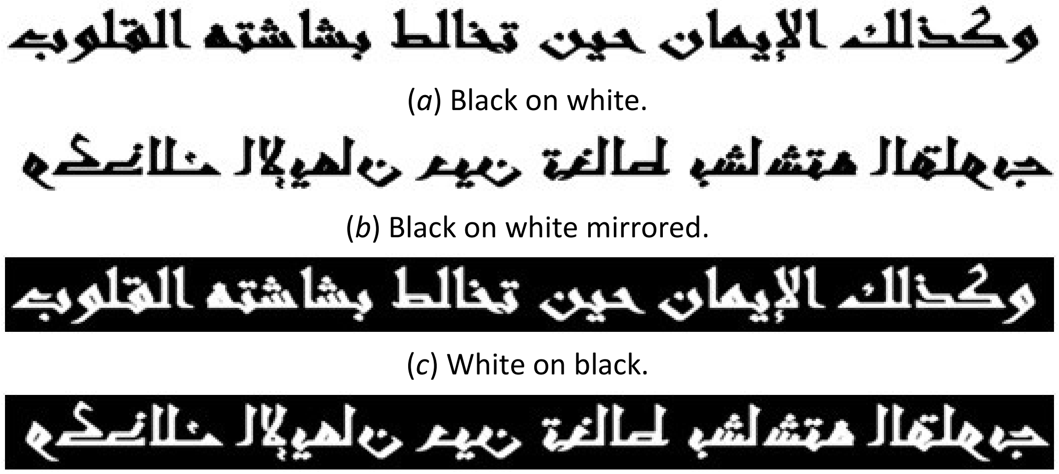


Figure 4.4: Basic definitions used in feature extraction.

4.4 Extraction Scheme with Thirty Features

In all our experiments, the extraction algorithm works on inverted text line normalized images. For Arabic the text line images were also mirrored (horizontally flipped) to ensure consistency with the algorithm. Figure 4.5 illustrates the mirroring and negation concepts. The mirroring is used to ensure compatibility with the left-to-write programming languages and tools that works with other left-to-right languages.



(*a*) Black on white.

(*b*) Black on white mirrored.

(*c*) White on black.

(*d*) White on black mirrored.

Figure 4.5: Arabic line image sample.

In this implementation, the width *p* of the horizontal sliding window is three pixels. The overlapping value *q* is one pixel. The window W_H slides from the left of the text line image till the right of the image (See Figure 4.6). Arabic text images are mirrored before the process. Each window W_H is divided into fifteen non-overlapping equal-height vertical areas (W_V) each with width of three pixels. The count of pixels with a value of 1 in each area is saved as one feature of the current sliding window. This actually counts the number of pixels with white intensity in the black and white text image. This will produce 15 features. Feature 16 is simply the count of pixels with value “1” for the whole of the sliding window. The remaining features, i.e. features 17 to 30, represents the count of pixels with value “1” for each two consecutive areas starting from area 1.

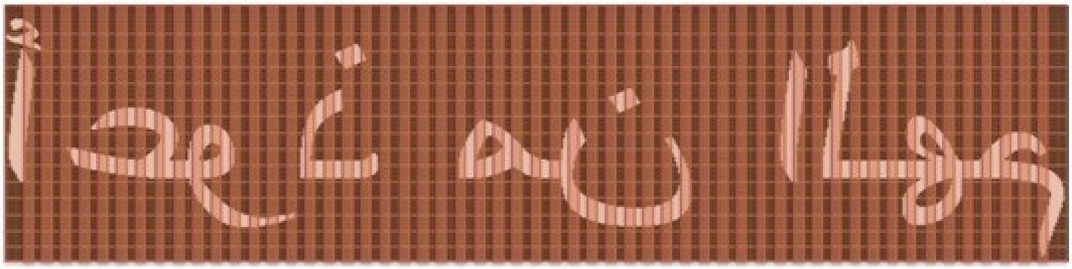


Figure 4.6: Horizontally sliding windows (W_H).

Figure 4.7 shows visually how the algorithm works. The windows W1, W2..., W6 are presented for illustration purposes only. They are instances of $W_{H.}$ A sliding window (W_H) is represented by W1 in the figure. W2 and W3 represent two consecutive overlapping instances of the suggested sliding window. W4 of Figure 4.7 shows the fifteen non-overlapping areas (W_V) of an instance of a sliding window where the first fifteen features are taken by counting the number of ones in each area. W5 shows a whole sliding window where feature 16 is computed by counting the number of ones in the window. W6 of Figure 4.7 shows the remaining fourteen features. Again each feature is simply the count of ones in each consecutive two areas. An overlapping area is always assumed in these features. Feature 17 is the count of ones in areas 1 and 2. Feature 18 is the count of ones in areas 2 and 3. Feature 19 is the count of ones in areas 3 and 4, and so on. Feature 30 is the count of ones in areas 14 and 15. The feature vector of the line image represents the matrix that contains the values of the thirty features for each sliding window.

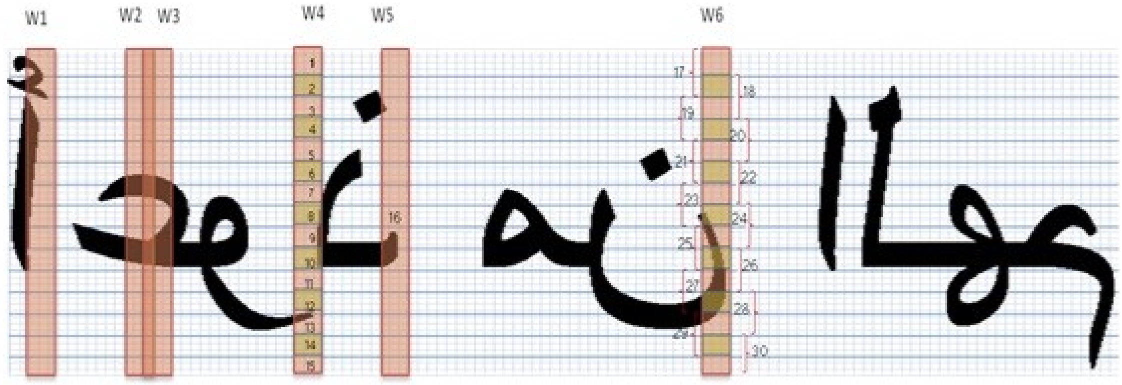


Figure 4.7: Proposed density features.

It is worth mentioning that this feature extraction scheme is language dependent. It should be fine-tuned for different languages. Fine-tuning could be done in different ways by adding and/or removing some grouping of the main fifteen suggested areas.

4.5 Extraction Scheme with Sixteen Features

In this implementation, the image line is divided into the eight main areas that govern letters' individualities. Figure 4.8 shows those areas. Table 4-1 illustrates the features and windows used in this feature extraction implementation.

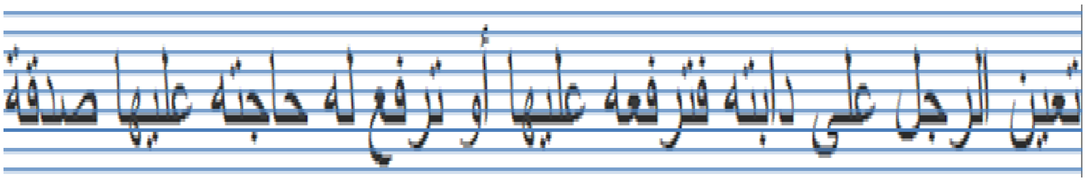

تعين الرجل على دابته فترفعه عليها أو ترفع له حاجته عليها صدقة

Figure 4.8: Eight main areas used for feature extraction visualized on an image line.

Table 4-1: Features and windows used in the 16-feature extraction case.

Features F_{16}	Features F_{15}	Features F_3 to F_4	Features F_9 to F_{12}	Features F_1 to F_8
$F_{16} = F_{13} + F_{14}$		$F_{14} = F_{11} + F_{12}$	$F_{12} = F_7 + F_8$	F_8 (sum of white pixels in 8th W_V)
				F_7 (sum of white pixels in 7th W_V)
	$F_{15} = F_{10} + F_{11}$		$F_{11} = F_5 + F_6$	F_6 (sum of white pixels in 6th W_V)
				F_5 (sum of white pixels in 5th W_V)
		$F_{13} = F_9 + F_{10}$	$F_{10} = F_3 + F_4$	F_4 (sum of white pixels in 4th W_V)
				F_3 (sum of white pixels in 3rd W_V)
			$F_9 = F_1 + F_2$	F_2 (sum of white pixels in 2nd W_V)
				F_1 (sum of white pixels in 1st W_V)

Starting from the first pixel of the text line image, a vertical segment (W_H) of 3 pixels width (p) and a height h equal to the height of the text line image is used. A window (W_V) of 3 pixels width and $h/8$ height is used to estimate the number of white pixels (as we are working on negated images) in the windows of the first level of the hierarchical structure. Eight vertically non-overlapping windows (W_V) are used to estimate the first 8 features (features 1 to 8). Four additional features (features 9 to 12) are estimated from four vertically non-overlapping windows of 3 pixels width and $h/4$ height (windows of the second level of the hierarchical structure). Then an overlapping window with 3 pixels width and $h/2$ height (windows of the third level of the hierarchical structure) with an overlap of $h/4$ is used to calculate three features (features 13 to 15). The last feature (feature 16) is found by estimating the number of white pixels (in a black background) in the vertical segment as a whole (the window of the fourth level of the hierarchical structure). Hence, 16 features were extracted for each horizontal window slide (W_H). To calculate the following features, the window (W_H) is moved horizontally, keeping an overlap of one pixel (the value of q). Sixteen features were extracted from each vertical strip and served as a feature vector in the training and/or testing processes.

4.6 Extraction Scheme with Ten Features

To be more practical, we present this implementation case by using a pseudo-code. Figure 4.9 shows the general structure of the algorithm in pseudo-code for possible implementation. As explained previously, the horizontal sliding window has a width of 3 pixels with 1 pixel overlapping. The strip represented by the window is divided into 8 equal non overlapping areas. Feature 1 is the count of white pixels in the first and the second areas. Feature 2 is the count of white pixels in the second and the third areas. Feature 3 is the count of white pixels in the third and the fourth areas. Feature 4 is the count of white pixels in the fourth and the fifth areas. Feature 5 is the count of white pixels in the fifth and the sixth areas. Feature 6 is the count of white pixels in the sixth and the seventh areas. Feature 7 is the sum of features 1, 2, and 3. Feature 8 is the sum of features 4, 5, 6. Feature 9 is the sum of features 2, 3, 4, and 5. The last feature is the count of white pixels in the whole of the sliding window. These 10 features are taken for each window along the width of the line image. Then, all features are grouped in a vector that represents the line image.

```
//  read the line image into a matrix with name lineImage;
    Part1Ends = LineImageHeight / 4;
    Part2Ends = LineImageHeight / 8;
    Part3Ends = LineImageHeight / 2;
    Part4Ends = LineImageHeight / 8;
    Part5Ends = LineImageHeight / 4;
    Part6Ends = LineImageHeight;
    m = 1; //counter for the horizontally sliding window
     for (k=1; k <= LineImageWidth - 2; k=k+2) {
            // Window's width is 3 & Overlap is 1
            Feature1(m) = sum(sum(lineImage(1:Part1Ends,k:k+2)));
            Feature2(m) = sum(sum(lineImage(Part1Ends+1:Part2Ends,k:k+2)));
            Feature3(m) = sum(sum(lineImage(Part2Ends+1:Part3Ends,k:k+2)));
            Feature4(m) = sum(sum(lineImage(Part3Ends+1:Part4Ends,k:k+2)));
            Feature5(m) = sum(sum(lineImage(Part4Ends+1:Part5Ends,k:k+2)));
            Feature6(m) = sum(sum(lineImage(Part5Ends+1:Part6Ends,k:k+2)));
            Feature7(m) = Feature1(m)+ Feature2(m)+Feature3(m);
            Feature8(m) = Feature4(m)+ Feature5(m)+Feature6(m);
            Feature9(m) =  Feature2(m)+Feature3(m)+Feature4(m)+ Feature5(m);
            Feature10(m) = Feature7(m)+Feature8(m);
        m=m+1;
       }    // end for k
        if (mod(LineImageWidth,2) == 0) { // Adjust for the last smaller window strip
         }
// line_vectors is the vector where the features are saved
          line_vectors = [Feature1 Feature2 ... Feature10];
```

Figure 4.9: Pseudo-code for a feature extraction algorithm.

4.7 Conclusion and Summary

Based on an analytical study of the individualities of Arabic alphabets, a technique based on the sliding window principle was implemented to extract text features. A window with variable width and height was used. Horizontal and vertical overlapping windows were investigated. In many experiments we tried different values for the window width, height, vertical overlapping, and horizontal overlapping. Then different types of windows were utilized to get more features from each vertical segment and to decide on the most proper window size and the number of overlapping cells vertically and horizontally. The direction of the text line images is considered as the feature extraction axis.

It has to be noted that the window size and vertical and horizontal overlapping are made settable. That is, the values of these parameters could be set and chosen to suit different feature extraction experiments. By setting the values of the size of the sliding window and the overlapping pixels a modified algorithm will be ready for testing. Hence different features may be extracted using different window sizes and vertical and horizontal overlapping.

Some of the advantages of the technique introduced in this chapter are: extracting a small number of one type of features (density); implementing different sizes of windows; using a hierarchical structure of windows for the same vertical strip; and applicability to other languages.

The next chapter will discuss the automatic recognition of printed Arabic text using the proposed feature extraction schemes introduced in this chapter.

Chapter 5. Training and Classification for Single Fonts

5.1 Introduction

After introducing the prepared data and the new proposed feature extraction schemes in the previous chapters, this chapter introduces the HMM techniques used to recognize Arabic text selected randomly from the prepared data using the new proposed feature extraction schemes.

The chapter presents the training and classification techniques used for Arabic printed text recognition. Section 5.2 briefly describes the HMM. The vector quantization process is explained in Section 5.3. Section 5.4 presents the language model used. Section 5.5 explains the methodology behind this research. The normalization process is discussed in Section 5.6. The procedure for selecting training and testing line images for experiments is presented in Section 5.7. Training related issues are discussed in Section 5.8. Single-font classifications are discussed in Section 5.9. Section 5.10 presents the summary of the chapter.

5.2 HMM

Several research papers have been published using HMM for text recognition. Examples of these papers are Khorsheed [*86*], Alma'adeed et al. [*142*], Bazzi et al. [*120*], Abbas et al. [*143*], Hu et al. [*8*], and Mohamed & Gader [*144*]. The use of HMM is very popular in speech recognition where the speech waveforms are computed as a function of an independent variable to formulate a sequence of vectors of discrete parameter. This is usually done by using sliding frames/windows. A similar technique is used in off-line text recognition where the independent variable is in the direction of the line length. See Bazzi et al. [*120*] and Khorsheed et al. [*70*].

In our experiments a left-to-right HMM is implemented for Arabic printed text recognition. Figure 5.1 displays the case of a 7-state HMM, showing that transition is allowed to the current, the next, and the following states only. This is in line with several research studies using HMM (Bazzi et al. [*120*] and [*46*]). This model, irrespective of the used number of states, allows relatively large variations in the horizontal position of the Arabic text. The sequence of state transition in the training and testing of the model is related to each text segment feature observations. That is, each shape of Arabic character is represented by an HMM with the used number of states, 7 states are used in Figure 5.1 as an example. Hence, the line image is represented by the composed HMM models that represents the images of the shapes of the characters in sequence.

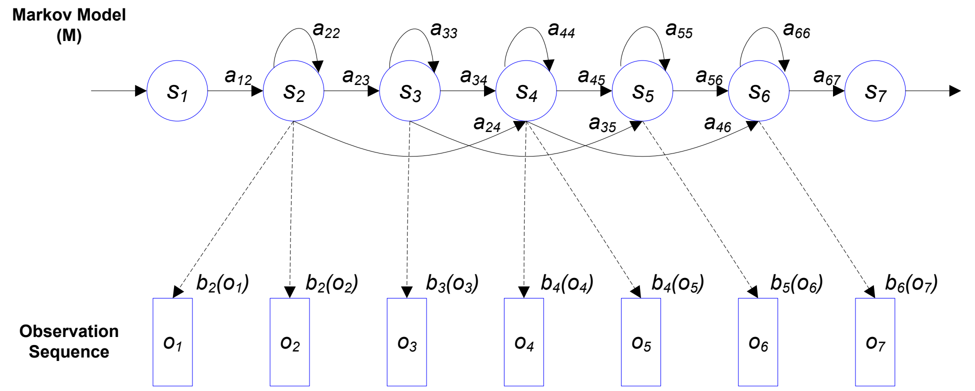


Figure 5.1: Seven-state HMM.

Each Arabic character image is represented by a sequence of character vectors or observations *O*, defined as

$$\boldsymbol{O} = \boldsymbol{o_1}, \boldsymbol{o_2}, \cdots, \boldsymbol{o_f} \qquad \textbf{(1)}$$

where o_f is the character vector observation at frame *f*. The character recognition problem can be regarded as that of computing

$$\underset{\boldsymbol{i}}{\mathbf{arg}\,\boldsymbol{max}} \quad \{\boldsymbol{P}(\boldsymbol{C_i}|\boldsymbol{O})\} \qquad \textbf{(2)}$$

where $C_i$is the *i*th character. This probability is computed using Bayes' Rule

$$\boldsymbol{P}(\boldsymbol{C_i}|\boldsymbol{O}) = \frac{\boldsymbol{P}(\boldsymbol{O}|\boldsymbol{C_i})\boldsymbol{P}(\boldsymbol{C_i})}{\boldsymbol{P}(\boldsymbol{O})} \qquad \textbf{(3)}$$

Thus for a given prior probabilities of a character $P(C_i)$, the most probable character depends only on the likelihood $P(O|C_i)$. Estimating the joint conditional probability $P(o_1, o_2, ..|C_i)$ directly seems to be impractical due to the dimensionality of the observation sequence *O*. However, such joint conditional probability could be estimated by using a parametric model such as Markov model. Hence, the dificulty with computing $P(O|C_i)$ is replaced by the problem of estimating Markov model parameters, which is a much simpler problem.

In hidden Markov models, it is assumed that the sequence of observed character vectors representing each character is generated by a Markov model similar to the one in Figure 5.1.

A Markov model is a finite state machine that changes its state at each time (frame) unit (*f*). With each change of state (moving from state *i* to state *j*) a character vector *O*$_f$ is generated from the probability density $b_j(o_f)$. Moreover, the transition from state *i* to state *j* is governed by the discrete probability a_{ij}. An example of this process is shown in Figure 5.1. The model in this example has 7 states where it moves (in this example) through the state sequence $S = 1, 2, 2, 3, 4, 4, 5, 6, 7$ to generate the sequence $o_1, o_2, \cdots, o_7$. The start state and the final state of this model are nonemitting states to allow building composed models. An example of composing three models of three character shapes each with 7 states is shown in Figure 5.2.

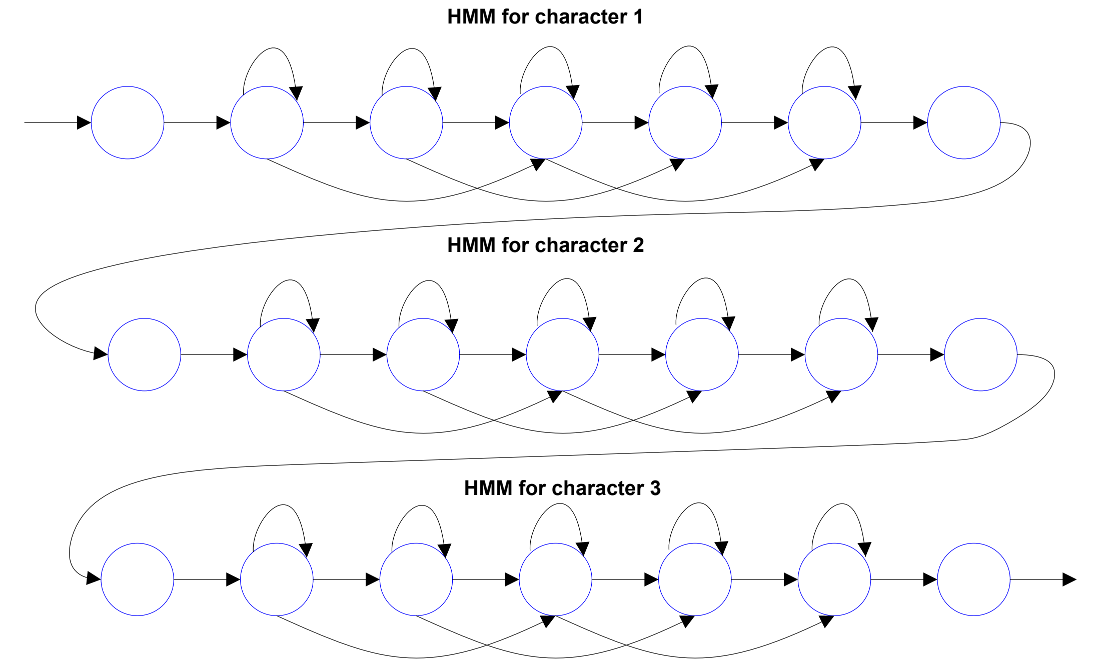


Figure 5.2: Example of composing 3 HMM models.

The probability of generating *O* by the model *M* through the state sequence *S*

$$\boldsymbol{P(O,S|M) = P(O|C_i)} \qquad \textbf{(4)}$$

is the product of the probabilities of the outputs and the probabilities of the transitions.

$$\boldsymbol{P(O|C_i) = a_{12}b_2(o_1)a_{22}b_2(o_2)a_{23}b_3(o_3)\cdots} \qquad \textbf{(5)}$$

However, the state sequence *S* is unknown and this is why this Markov model is called *Hidden Markov model*.

$P(O|C_i)$ which is now represented by $P(O|M)$ can be calculated as follows.

As the state sequence is unknown, the probability is computed by summing overall possible state sequences

$$\boldsymbol{S = s(1), s(2), s(3), \cdots, s(F)} \qquad \textbf{(6)}$$

$$\boldsymbol{P(O|M)} = \left\{\sum_{\boldsymbol{S}} \boldsymbol{a_{s(0)}a_{s(1)}} \prod_{\boldsymbol{f=1}}^{\boldsymbol{F}} \boldsymbol{b_{s(f)}\left(o_f\right)a_{s(f)s(f+1)}}\right\} \quad \textbf{(7)}$$

where $s(0)$ is the entry state and $s(F+1)$ is the exit state.

The latest equation could be approximated as

$$\boldsymbol{\widehat{P}(O|M)} = \underset{\boldsymbol{S}}{\boldsymbol{max}} \quad \left\{\boldsymbol{a_{s(0)}a_{s(1)}} \prod_{\boldsymbol{f=1}}^{\boldsymbol{F}} \boldsymbol{b_{s(f)}\left(o_f\right)a_{s(f)s(f+1)}}\right\} \quad \textbf{(8)}$$

This equation is usually computed by recursion with the assumption that the parameters a_{ij} and $b_j\left(o_f\right)$ are known for each model M_i.

The advantage of using HMM-based techniques is the by-product segmentation while doing the recognition. Although text is modelled as the composition of shapes of letters, HMMs avoid text pre-segmentation in both training and classification phases. Moreover, using HMMs allows dealing with variable-lengths sequences of observations [*145*]. Furthermore, given a sufficient number of representative training examples of each character, the parameters of the model can be determined by a re-estimation procedure. The model represents implicitly different sources of variations inherited in character vectors representing images of letters.

Figure 5.3 summarizes the use of HMM for character recognition. Using a set of examples of character images, a HMM is trained for that character. In this example only 3 characters were used. To recognize an unknown character, the likelihood of each model generating that character is captured.

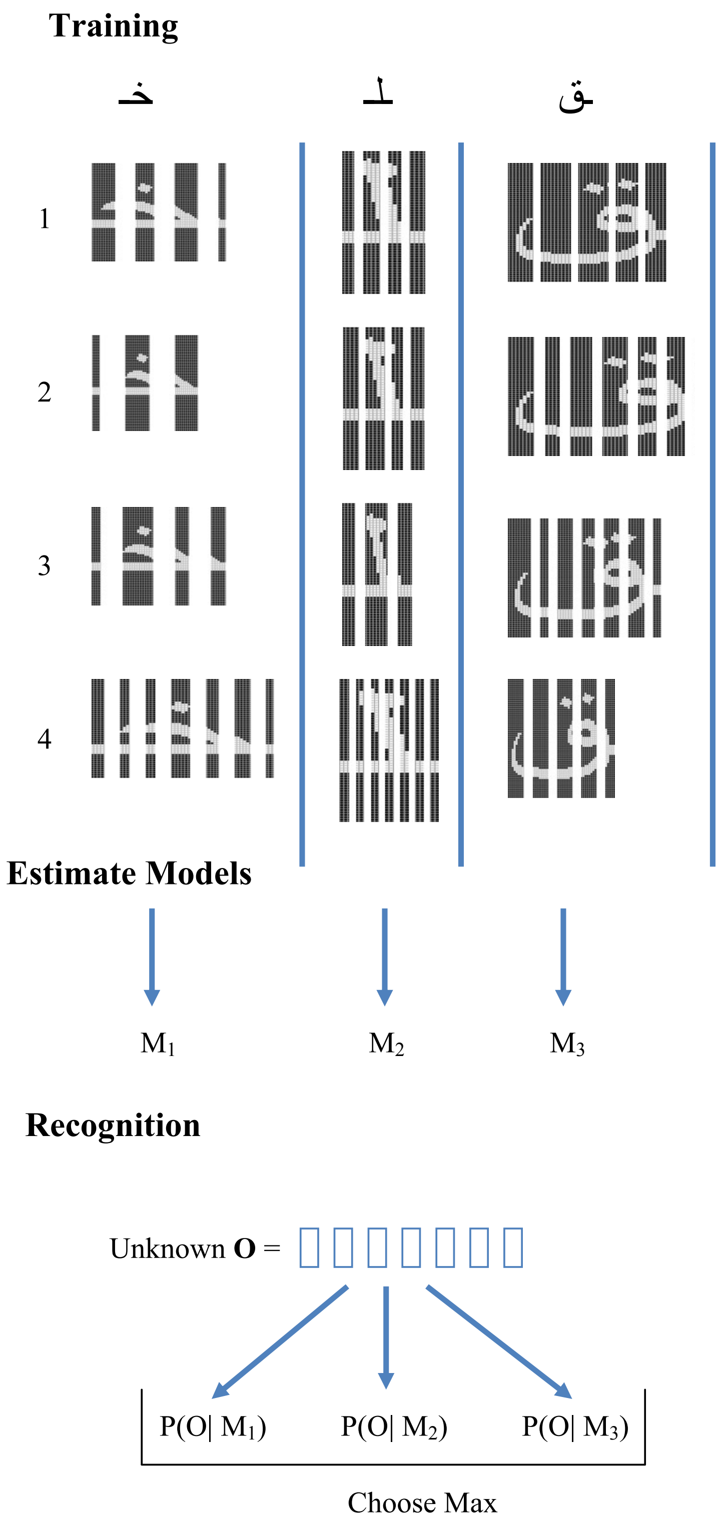


Figure 5.3: The use of HMM for character recognition.

5.3 Vector Quantization and Codebook

The feature vectors that represent the Arabic text line images are the training sequences for the prototype. These vectors are called source vectors. Each source vector consists of sequence of vectors, where each represents a line partition. Vector Quantization (VQ) is the process of clustering these consecutive sequences of partitions into encoding regions. A consecutive sequence of partitions appearing repeatedly in the source vectors is referred to as a codevector. Each encoding region is represented by a codevector. The set of all unique codevectors that represent encoding regions in the training sequence defines the codebook. Given any codevector, it should be represented by the nearest clustered encoding region (a codebook entry) that minimizes the distortion error.

5.4 The Bigram Language Model

The regularity in a natural language could be captured statistically by an N-gram statistical languages model [*146*], where N is the number of involved neighbours of the text of the language. The neighbours could be words, sub-words, or characters depending on the application. If N is 2, the model is called bigram model. A statistical language model has a lot of applications in natural language processing. Some of these applications are machine translation, spell checking, information retrieval, and data mining.

In our prototype the statistical language model we are using is the bigram of the shapes of Arabic letters. Simply, the bigram model of the shapes of Arabic letters captures the probability of a shape of a letter appearing after a given Arabic shape. This is why it is a bigram model and not a unigram or trigram model. Two Arabic letter shapes are involved in the statistics at each time. With the assumption that the probability of the current Arabic letter shape depends only on the previous Arabic letter shape. This probability is used in training and recognition to help in deciding the right class (shape). The bigram probability is computed as the number of times the current Arabic shape appears in the text after a given previous Arabic letter shape over the total number of appearances for the current Arabic shape.

5.5 Methodology

Figure 5.4 shows the block diagram for the system prototype. After preparing the text images and their labelling (see sections 3.5, 3.6 and 3.7), the pages are segmented into line images, which are converted to black and white images. The line images are normalized to have equal heights. Line widths vary depending on the original length of the lines. Then the features are extracted. A file that contains the feature vectors of each line was prepared. The feature vector contains the features extracted for each vertical strip of the image of the text line by one of the three methods described earlier (see sections 4.4, 4.5, 4.6). All feature vectors of the vertical strips of the line image are represented in a 2-D matrix. The list of matrices representing all lines for training are passed for vector quantization to cluster the features streams (matrices) into clusters represented in a one one-dimensional vector (codebook). This codebook is used to convert the feature stream of the image line into discrete observations that could be used to generate HMM models. The observations are passed to the training module along with the ground truth text and the statistical analysis results of the ground truth text that represents the language model. The training module generated the parameters of the HMM model for each shape of each Arabic character.

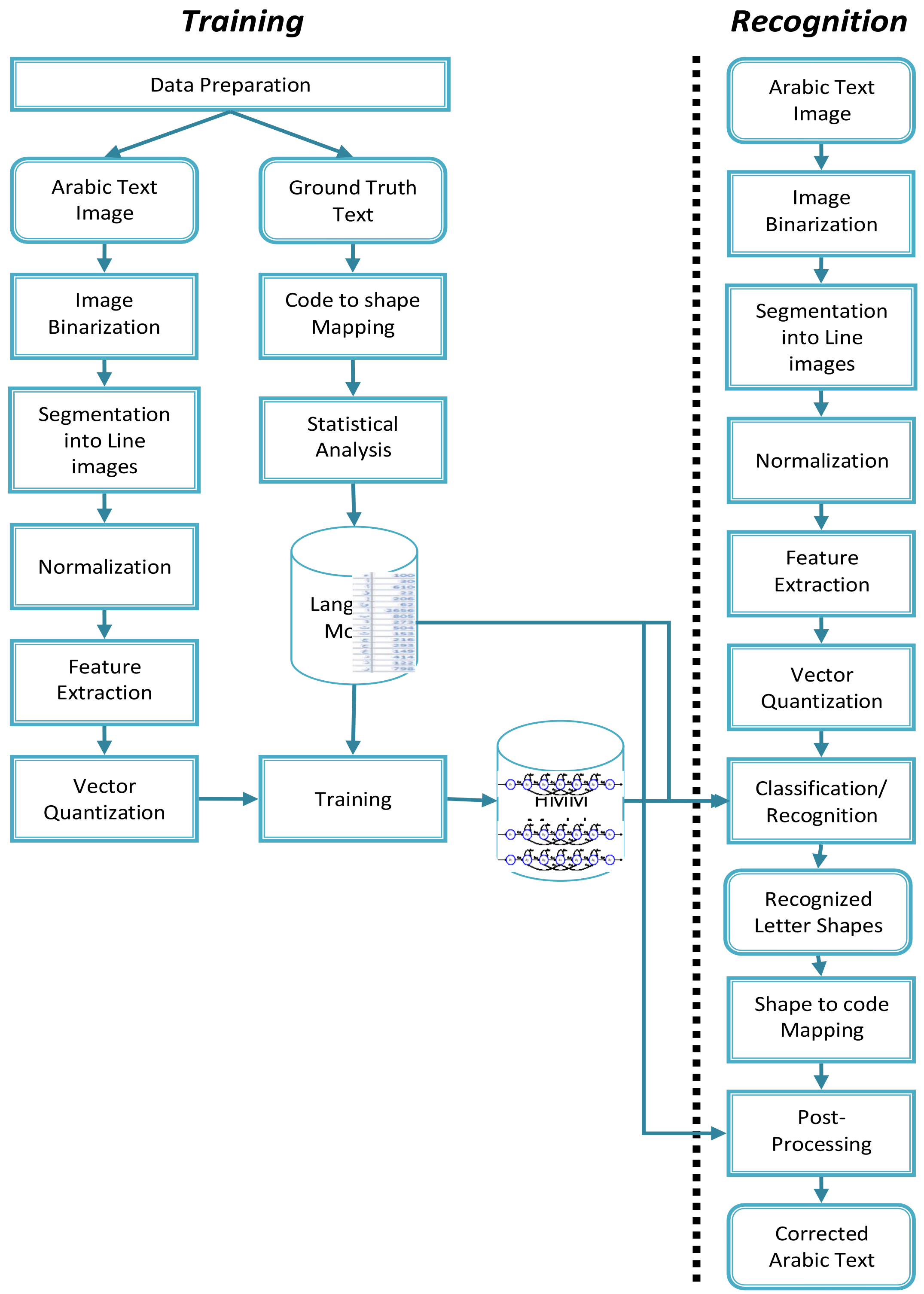


Figure 5.4: Printed Arabic text recognition block diagram.

In the classification stage, a similar process is followed. The features of the normalized line images are extracted and changed to discrete observations through the quantized vector. The observations are classified to fit the most suitable character-shape model. The corresponding class (shape) is reported. The shapes are remapped to their corresponding characters. The recognized text is processed by the post-processing module for possible corrections. The corrected text is the output of the prototype.

5.5.1 Extracting Features

We extract the features of Arabic text line images by using the sliding window principle to calculate the features based on a sliding vertical strip which covers parts of the character. However, our technique differs from the general trend of other researchers. We implement a hierarchical window structure with different window sizes and horizontal and vertical overlapping. In addition, we extract only a limited number of simple features of one type per vertical strip. We have successfully used 10 features, 16 features, and 30 features of one type compared to 80 features of four types of features used by Bazzi et al. [*120*] and [*46*]. The results using the sixteen features have been reported for other researchers [*147*]. We bypass the need for segmenting Arabic characters, and our technique is applicable to other languages [*148*].

We have investigated using different numbers of states and codebook sizes, and selected the best performing ones. Although each character model could have a different number of states, we decided to adopt the same number of states for all characters in a font. However, the number of states and codebook sizes for each font, in relation to the best recognition rates for each font, are different.

5.5.2 HMM Toolkit (HTK)

We use the same HMM classifier without modification as implemented in the Hidden Markov Model toolkit (HTK) [*149*]. However, we implement our own parameters to tune the HMM. We allowed transition to the current, the next, and the following states only. This structure allows nonlinear variations in the horizontal position. HTK models the feature vector with multiple Gaussian functions called mixture of Gaussians or Gaussian Mixture. It uses the Viterbi algorithm in the recognition phase which searches for the most likely sequence of a character given the input feature vector.

5.6 Normalization of Line Images

When experimenting with single fonts line image, we have noticed that normalization has no major effects on the accuracy of the recognition. The reason is that in single font recognition, the original line images of the same font have the same height. The effect of normalization appears clearly when multi-font experiments are considered. Original line images were prepared with some blank pixels around the line image. Cropping these blank pixels from around the line was also considered. Different types of line image normalizations were tried with and without cropping of blank pixels. We have experimented with different height normalizations. We have run experiments using 60 pixels, 80 pixels, 100 pixels, 120 pixels, 150 pixels, and 180 pixels. The data which we are using consist of text written using 18 points font size. Although we used the same size for all fonts, their actual image sizes were not consistent. Table 5-1 shows the height of each line image for different fonts before and after line image cropping.

Table 5-1: Line image heights for the fonts in use.

Font	Original Height in Pixels	Height after Cropping in Pixels
Akhbar	77	45 - 52
Andalus	77	42 - 54
Arial	57	50 - 54
Naskh	105	62 - 72
Simplified	83	50 -58
Tahoma	60	54 - 60
Thuluth	103	58 - 77
Traditional	75	47 - 55

5.7 Training and Testing Sets

In order to have enough samples of each font class, two datasets were used for the training and testing phases. These datasets are PATS-A01 and PATS-A02 (see Section 3.5). The first set consists of a total of 2766 line images and the second set consists of a total of 318 line images. From the first set 2500 line images were used for training and the remaining 266 line images were used for testing. From the second set 286 line images were used for training and the remaining 32 line images for testing. There is no overlap between the training and testing samples. For each dataset, nine different sets were prepared for training and testing (actually ten for dataset PATS-A01 and nine for dataset PATS-A02). In each training and testing set the test line images were selected using a random number generator. Then, the remaining unselected line images were included in the training set. This procedure was repeated 9 times for both datasets PATS-A01 and PATS-A02. In our experiments, we used these nine training and testing sets of both datasets for each font we have used. The files of these training sets for both datasets are provided in the enclosed CD-ROM (See Appendix A). It is worth mentioning that each training set contains enough samples of all letter shapes as the database is large enough to afford this.

5.8 Training

A large number of trials were conducted to find the most suitable combinations of the number of suitable states and codebook sizes. Different combinations were tested. The states that were experimented with ranged from 3 to 15. The sizes of codebooks that were experimented with were 32, 64, 128, 192, 256, 320, 384, 512, and in between them (See Section 5.3).

It has been noticed that the larger the size of the codebook the better the performance for a given number of states. Similar findings were reported by several researchers including Zhang et al. [*150*], Al-Ma'adeed [*151*], and El-Mahallawy [*152*]. However, the size of the codebook is limited to the maximum clustering regions that could be generated from the codevectors. Hence, the size and the variation in the training samples play a major role in limiting the highest size of the codebook. Moreover, more computation time is expected when a large codebook is used.

When the number of states is considered, ideally, the suitable number depends on the shape of the letter. Some letters have more shapes than others and, hence would require more states. However, because of the nature of the HMM, a single HMM with a fixed number of states could be used for all shapes. The model allows transitions to the same state as well as to jump to the state after the next state; see Figure 5.1. This accommodates for both wide and narrow shapes of letters.

5.8.1 Performance Measures

Two performance measures were used to evaluate the efficiency of the algorithms used: correctness and accuracy. The following two equations define these measures.

$$\boldsymbol{Correctness}\% = (\boldsymbol{samples} - (\boldsymbol{substitutions} + \boldsymbol{deletions}))/\boldsymbol{samples} \times \mathbf{100} \qquad \textbf{(9)}$$

$$\boldsymbol{Accuracy}\% = (\boldsymbol{samples} - (\boldsymbol{substitutions} + \boldsymbol{insertions} + \boldsymbol{deletions}))/\boldsymbol{samples} \times \mathbf{100} \qquad \textbf{(10)}$$

Word Error Rate (WER) percentage is calculated as

$$\boldsymbol{WER}\% = (\mathbf{100} - \boldsymbol{Correctness}) \times \mathbf{100} \qquad \textbf{(11)}$$

Figure 5.5, Figure 5.6, Figure 5.7, and Figure 5.8 show graphs of the percentage of correctness versus used number of states for some experiments with dataset PATS-A02 for all used eight fonts using a single HHM. These figures are samples of the nine different sets for training and testing. All results of the training and testing sets were in the same ranges. Figure 5.9 shows the percentage correctness versus the number of states for all the nine training and testing sets for all the eight fonts used.

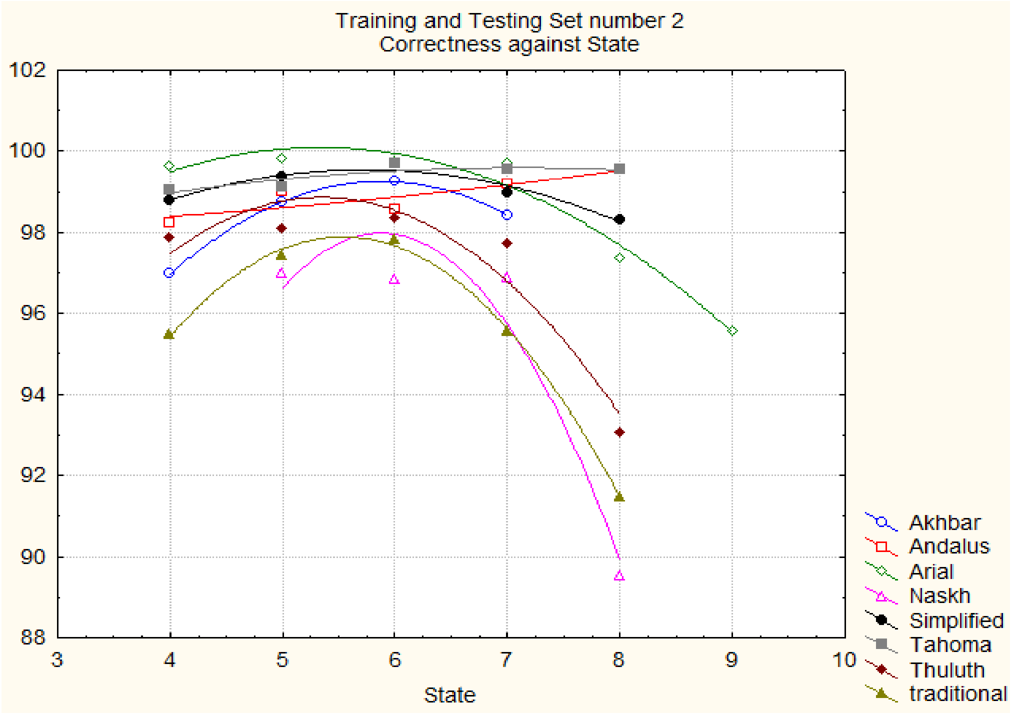


Figure 5.5: Correctness versus number of states for training and testing Set 2 for dataset PATS-A02.

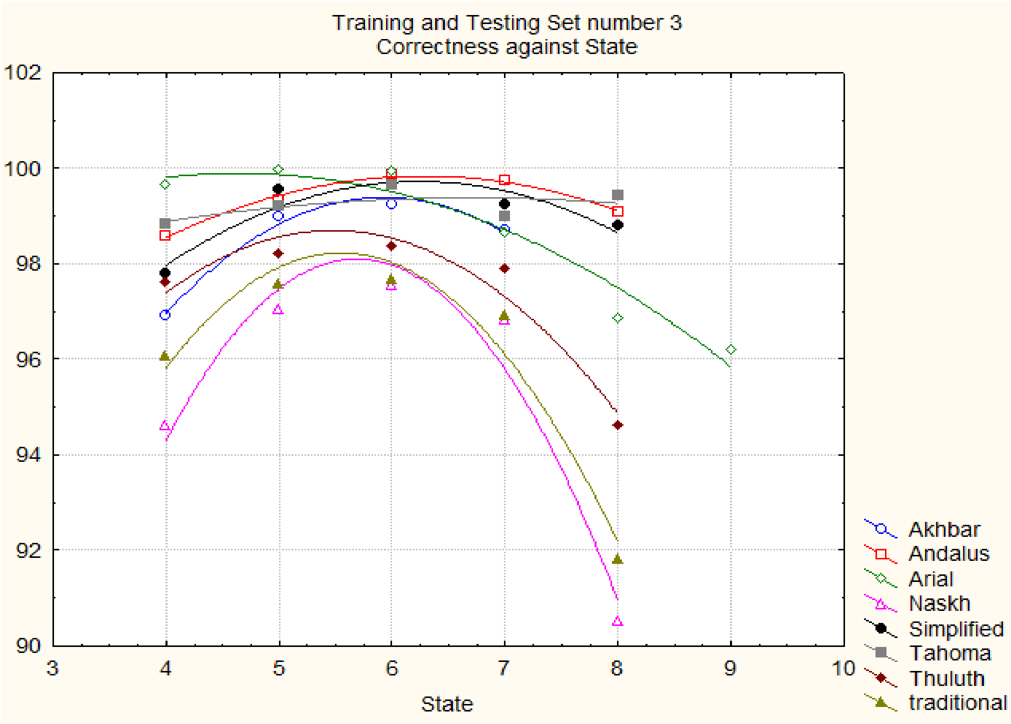


Figure 5.6: Correctness versus number of states for training and testing Set 3 for dataset PATS-A02.

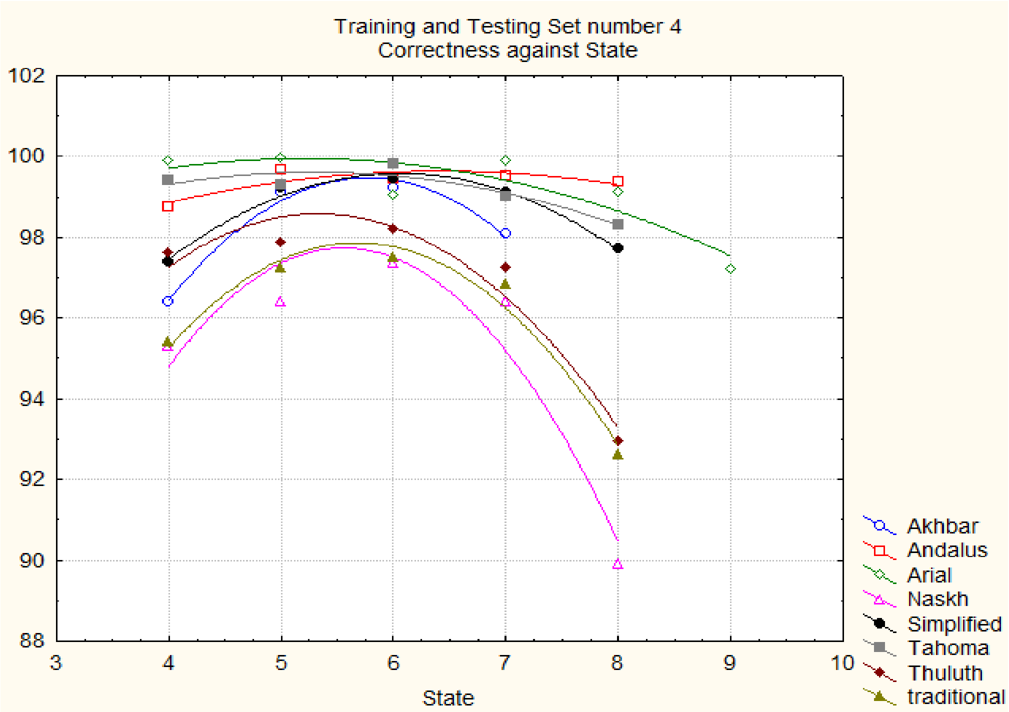


Figure 5.7: Correctness versus number of states for training and testing Set 4 for dataset PATS-A02.

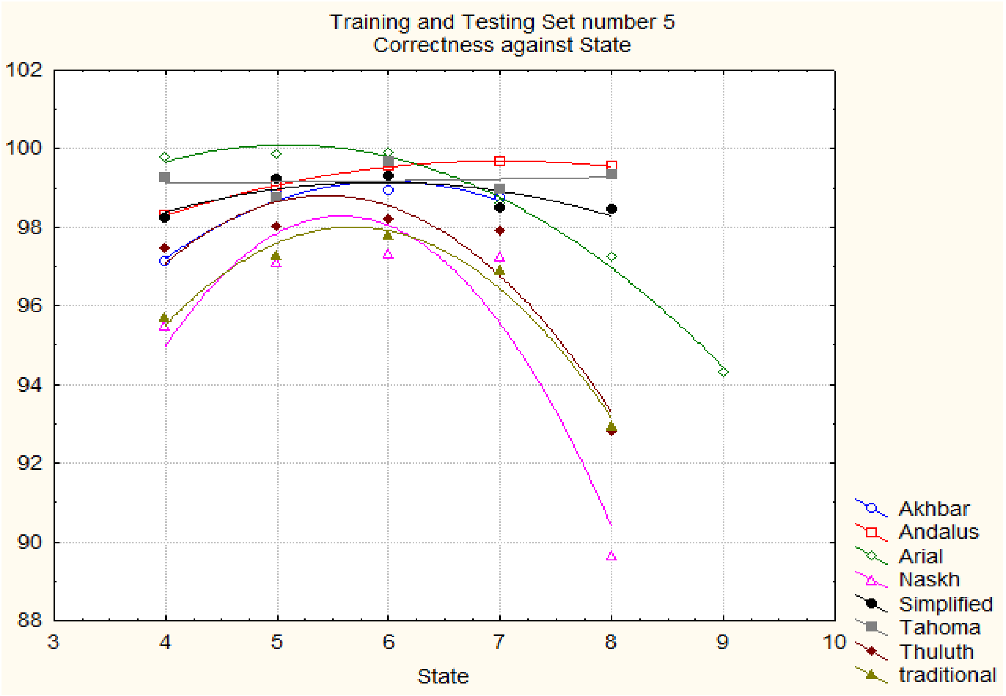


Figure 5.8: Correctness versus number of states for training and testing Set 5 for dataset PATS-A02.

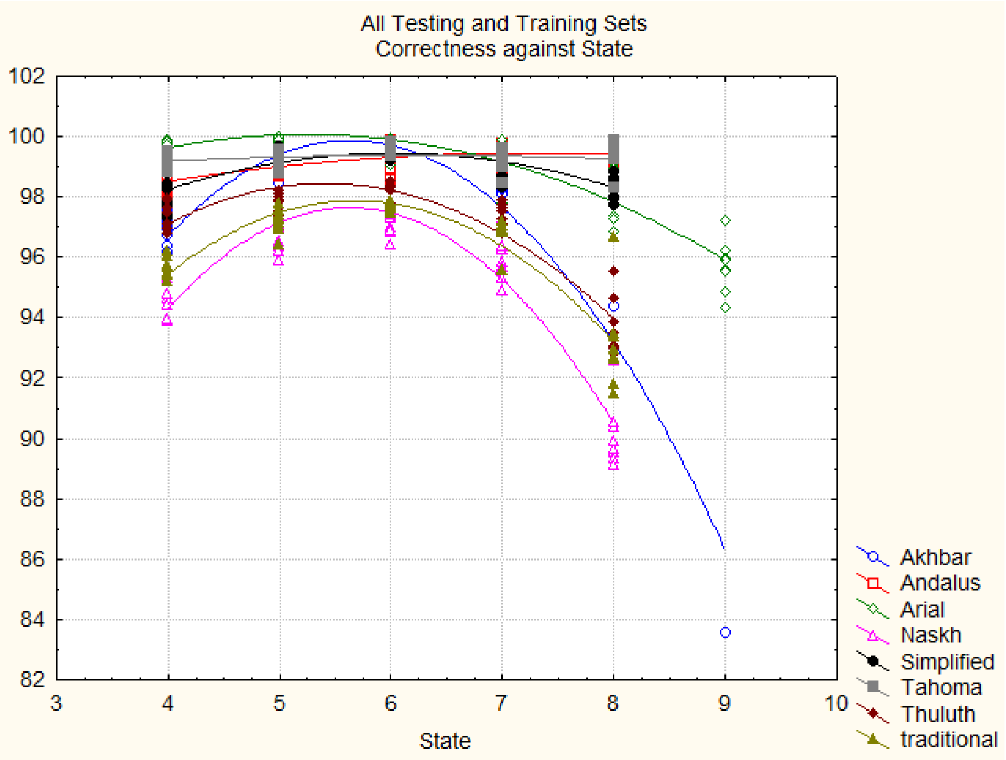


Figure 5.9: Correctness versus number of states for training and testing of the nine sets for dataset PATS-A02.

Table 5-2: Combinations of number of states and size of codebook used for different fonts for dataset PATS-A01.Table 5-2 shows the best combinations, which were found experimentally, and provide the best recognition rates (accuracy and correctness) for each font in dataset PATS-A01. The best combinations of codebook size and number of states for each font in dataset PATS-A02 is shown in Table 5-3. It is expected to be slightly different as the dataset PATS-A02 is less than one eighth the size of the dataset PATS-A01. However, it is still a good representative of the language as it is covering adequate samples of all the basic shapes of Arabic letters (see Section 3.5) and the recognition rate after training is found to be high.

Table 5-2: Combinations of number of states and size of codebook used for different fonts for dataset PATS-A01.

Font Name	Number of Sates	Codebook size
Arial	5	256
Tahoma	7	128
Akhbar	5	256
Thuluth	7	128
Naskh	7	128
Simplified Arabic	7	128
Traditional Arabic	7	256
Andalus	7	256

Table 5-3: Combinations of number of states and size of codebook used for different fonts for dataset PATS-A02.

Font Name	Number of Sates	Codebook size
Arial	5	192
Tahoma	8	128
Akhbar	6	80
Thuluth	6	96
Naskh	6	56
Simplified Arabic	6	96
Traditional Arabic	6	80
Andalus	6	88

5.9 Classification

The results of testing 266 lines using dataset PATS-A01 are summarized in Table 5-4. The results are the averages of the results of the nine testing and training sets for each font. Actually, the result for each font for each training and testing set is the averages of the recognition rate of each shape involved in the testing. The table also shows the effect of having a unique code for each shape of each character in the classification phase (Columns 2 & 3) and then combining the shapes of the same character into one code (Columns 4 & 5). In all cases there are improvements in both correctness and accuracy in combining the different shapes of the character after recognition into one code. This is expected and justifiable. When a shape *X* is misrecognized as *Y* (not recognised correctly), the features of *Y* are nearly similar to the features of *X*. So it is most probable that the shapes *X* and *Y* are different shapes of the same letter. Different shapes of the same letter have, in many cases, semi-similar features. That is, their codevectors belong to nearer clusters.

Table 5-4: Summary of results per font type with and without shape expansion for dataset PATS-A01 of all training and testing sets.

	With expanded shapes		With collapsed shapes	
Text font	**Correctness %**	**Accuracy %**	**Correctness %**	**Accuracy %**
Arial	99.89	99.85	99.94	99.90
Tahoma	99.80	99.57	99.92	99.68
Akhbar	99.33	99.25	99.43	99.34
Thuluth	98.08	98.02	98.85	98.78
Naskh	98.12	98.02	98.19	98.09
Simplified Arabic	99.69	99.55	99.84	99.70
Traditional Arabic	98.85	98.81	98.87	98.83
Andalus	98.92	96.83	99.99	97.86

To calculate the average correctness and accuracy percentages for each font and for each testing experiment, the resultant confusion matrix for these runs is analyzed. The matrix is too large to be displayed in raw format, as it consists of 126 rows by 126 columns. The confusion matrix is represented in a more informative way by collapsing all different shapes of the same character into one entry and by listing error details for each character. This will actually be the result after converting the recognized text from the unique coding of each shape to the unique coding of each character. This is done by the contextual analysis module (Shape to Code Mapping model), a tool we built for this purpose. Table 5-5 shows an example of a partial confusion matrix. The table has only 17 shapes representing 8 characters.

The following subsections discuss the classification results for the fonts used (Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic) and for several combinations of these fonts.

Table 5-5: Partial confusion matrix.

	ء	آ	أ	أ	ؤ	إ	إ	ئ	ئ	ئ	ئ	ا	ا	ب	ب	ب	ب
ء	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
آ	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
أ	0	0	415	0	0	0	0	0	0	0	0	0	0	0	0	0	0
أ	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	0
ؤ	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0
إ	0	0	0	0	0	142	0	0	0	0	0	0	0	0	0	0	0
إ	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0
ئ	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ئ	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
ئ	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
ئ	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0
ا	0	0	0	0	0	0	0	0	0	0	0	1124	0	0	0	0	0
ا	0	0	0	0	0	0	0	0	0	0	0	0	988	0	0	0	0
ب	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0
ب	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0
ب	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	288	0
ب	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	352

5.9.1 Single Fonts

The dataset PATS-A01 was used for all the experiments reported in this sub-section.

5.9.1.1 Classification of the Akhbar Font Text

Table 5-6 shows the classification results for the Akhbar font. The correctness for this font was 99.43% and the accuracy reached 99.34%. Seven letters and two ligatures had 45 substitutions plus 19 insertions. Twenty one substitutions were related to the ligature لم (See Table 5-7 for the shape of this ligature) which was confused with Meem م as Lam ل is very small in width. This resulted in 19 insertions to substitute for the errors.

Table 5-6: Classification results for Akhbar font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	80	80	0	100.0	0.0	3	0	96.3	96.3	
آ	10	6	4	60.0	40.0	0	0	60.0	60.0	
أ	483	482	1	99.8	0.2	1	0	99.6	99.6	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2108	2099	9	99.6	0.4	11	1	99.1	99.0	2ـب 4ـد 1ـص 1ـف 1ـك
ب	411	411	0	100.0	0.0	3	0	99.3	99.3	
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	420	0	100.0	0.0	0	0	100.0	100.0	
ث	122	122	0	100.0	0.0	2	7	98.4	92.6	
ج	170	170	0	100.0	0.0	0	0	100.0	100.0	
ح	234	234	0	100.0	0.0	0	0	100.0	100.0	
خ	113	113	0	100.0	0.0	0	0	100.0	100.0	
د	344	344	0	100.0	0.0	0	1	100.0	99.7	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	702	0	100.0	0.0	0	0	100.0	100.0	
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	639	639	0	100.0	0.0	1	6	99.8	98.9	
ش	119	118	1	99.2	0.8	0	0	99.2	99.2	1ـت
ص	415	415	0	100.0	0.0	0	0	100.0	100.0	
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	68	0	100.0	0.0	0	0	100.0	100.0	
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	816	2	99.8	0.2	0	0	99.8	99.8	1ـد 1ـلم
غ	44	43	1	97.7	2.3	0	0	97.7	97.7	1ـن
ف	493	493	0	100.0	0.0	2	0	99.6	99.6	
ق	465	462	3	99.4	0.6	2	0	98.9	98.9	3ـف
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	1441	1436	5	99.7	0.3	23	3	98.1	97.9	5ـلم
م	670	670	0	100.0	0.0	1	0	99.9	99.9	
ن	1018	1017	1	99.9	0.1	5	1	99.4	99.3	1ـل
ه	663	663	0	100.0	0.0	2	0	99.7	99.7	
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	40	0	100.0	0.0	0	0	100.0	100.0	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	0	100.0	100.0	
ى	86	86	0	100.0	0.0	0	0	100.0	100.0	
ي	1151	1140	11	99.0	1.0	8	0	98.4	98.4	11ـب
Blnk	4636	4636	0	100.0	0.0	1	0	100.0	100.0	
ﷲ	491	490	1	99.8	0.2	0	0	99.8	99.8	1ـه
لم	334	314	20	94.0	6.0	0	0	94.0		1ـص 19ـم
لى	327	327	0	100.0	0.0	0	0			
Ins			19							1ـا 7ـث 1ـد 6ـس 3ـل 1ـن

5.9.1.2 Classification of Andalus Font Text

Andalus font seems to be the most unambiguous font. Table 5-8 shows the classification results for this font. The correctness percentage was 99.99 and the accuracy percentage was 97.86. Using one code for the different shapes of a character after recognition improves the recognition rate for single fonts. This font (Andalus) shows the highest improvement of the recognition rate compared with all the other fonts used. Eleven letters out of 43 had some errors. Only two letters had actual substitutions. There were also 3 deletion instances. Most of the errors appearing in the accuracy percentage are artificial due to the use of the ligature ﷲ. It caused 476 insertion of the letter *Lam* ل. Removing this ligature from the analysis (as it should not be considered as a ligature in this font, see its shape in Table 5-7), will raise the accuracy to more than 99.6%. This font is suitable for automatic recognition of car plates containing Arabic characters. When assigning letters and numbers to car plates, one shape is used for all of the characters that have the same basic shape. Moreover, isolated characters and digits are used. This might lead to an accuracy reaching 100% neglecting the effect of noise.

Table 5-7: Ligatures لله and لم in different fonts.

Font Name	The ligature لله	The ligature لم
Arial	لله	لم
Tahoma	لله	لم
Akhbar	لله	لم
Thuluth	لله	لم
Naskh	لله	لم
Simplified Arabic	لله	لم
Traditional Arabic	لله	لم
Andalus	لله	لم

5.9.1.3 *Classification of Arial Font Text*

Table 5-9 shows the classifications results for the Arial font. The correctness percentage was 99.94 and the accuracy percentage was 99.90. Only four letters out of 43 had some errors. The letter ح has been substituted with the letter ج four times out of 234 instances. The only difference between the two characters is the dot in the body of the letter ج. The second error consists of two replacements of the letter ه with the letter ء out of 665 instances. The third error was substituting the ligature لأ with a blank four times out of 40. The fourth error was substituting the ligature لله once with ه out of 491 times. Other than the substitutions, 10 insertions were added (two of them were blanks). The blank problems were reported by several researchers including Bazzi [*120*].

5.9.1.4 *Classification of Naskh Font Text*

The classification results of the Naskh font are shown in compressed form in Table 5-10. The percentage of correctness is 98.19 and the accuracy percentage was 98.09.

There were around 200 substitutions and 21 insertions. The argument presented in the Thuluth font classification is also valid here as this font is tightly cursive and has a lot of overlapping. This font received the highest number of deletions among all other fonts. It has around 200 cases of deletions; half of them were for letters Meem م and Lam ل due to the ligatures used.

Table 5-8: Classification results for Andalus font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	83	83	0	100.0	0.0	0	0	100.0	100.0	
آ	10	10	0	100.0	0.0	0	0	100.0	100.0	
أ	484	484	0	100.0	0.0	0	0	100.0	100.0	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2119	2119	0	100.0	0.0	0	0	100.0	100.0	
ب	409	409	0	100.0	0.0	0	1	100.0	99.8	
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	420	0	100.0	0.0	0	0	100.0	100.0	
ث	124	124	0	100.0	0.0	0	0	100.0	100.0	
ج	170	170	0	100.0	0.0	0	0	100.0	100.0	
ح	234	234	0	100.0	0.0	0	0	100.0	100.0	
خ	113	113	0	100.0	0.0	0	0	100.0	100.0	
د	344	344	0	100.0	0.0	0	1	100.0	99.7	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	702	0	100.0	0.0	0	0	100.0	100.0	
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	640	0	100.0	0.0	0	0	100.0	100.0	
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	415	413	0	99.5	0.0	0	0	100.0	100.0	2ـض
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	68	0	100.0	0.0	0	0	100.0	100.0	
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	818	0	100.0	0.0	0	0	100.0	100.0	
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	495	495	0	100.0	0.0	0	0	100.0	100.0	
ق	467	467	0	100.0	0.0	0	0	100.0	100.0	
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2136	2136	0	100.0	0.0	0	476	100.0	77.7	
م	1005	1005	0	100.0	0.0	0	0	100.0	100.0	
ن	1023	1023	0	100.0	0.0	0	0	100.0	100.0	
ه	665	665	0	100.0	0.0	0	0	100.0	100.0	
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	40	0	100.0	0.0	0	0	100.0	100.0	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	206	206	0	100.0	0.0	0	0	100.0	100.0	
ى	413	413	0	100.0	0.0	0	0	100.0	100.0	
ي	1159	1158	0	99.9	0.0	0	0	100.0	100.0	1ـى
Blnk	4634	4634	0	100.0	0.0	3	0	99.9	99.9	3حذف
الله	491	253	0	51.5	0.0	0	0	100.0	100.0	238ـه
Ins			478							1ـب 1ـد 476ـل

Table 5-9: Classification results for Arial font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	83	83	0	100.0	0.0	0	0	100.0	100.0	
آ	10	8	2	80.0	20.0	0	0	80.0	80.0	
أ	484	484	0	100.0	0.0	0	0	100.0	100.0	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2114	2114	0	100.0	0.0	0	1	100.0	100.0	
ب	409	409	0	100.0	0.0	0	0	100.0	100.0	
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	420	0	100.0	0.0	0	0	100.0	100.0	
ث	124	124	0	100.0	0.0	0	0	100.0	100.0	
ج	170	170	0	100.0	0.0	0	0	100.0	100.0	
ح	234	230	4	98.3	1.7	0	0	98.3	98.3	ج4
خ	113	113	0	100.0	0.0	0	0	100.0	100.0	
د	344	344	0	100.0	0.0	0	1	100.0	99.7	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	702	0	100.0	0.0	0	0	100.0	100.0	
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	640	0	100.0	0.0	0	0	100.0	100.0	
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	415	415	0	100.0	0.0	0	0	100.0	100.0	
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	68	0	100.0	0.0	0	0	100.0	100.0	
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	818	0	100.0	0.0	0	0	100.0	100.0	
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	495	495	0	100.0	0.0	0	0	100.0	100.0	
ق	467	467	0	100.0	0.0	0	0	100.0	100.0	
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2136	2136	0	100.0	0.0	0	2	100.0	99.9	
م	1005	1005	0	100.0	0.0	0	0	100.0	100.0	
ن	1023	1023	0	100.0	0.0	0	0	100.0	100.0	
ه	665	663	2	99.7	0.3	0	0	99.7	99.7	ء-2
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	36	4	90.0	10.0	0	0	90.0	90.0	-4Blnk
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	4	100.0	98.1	
ى	413	413	0	100.0	0.0	0	0	100.0	100.0	
ي	1159	1159	0	100.0	0.0	0	0	100.0	100.0	
Blnk	4637	4637	0	100.0	0.0	0	2	100.0	100.0	
ﷲ	491	490	1			0	0	99.8	99.8	ـه1
Ins			10							-2 ـلا4 ـل2 ـد1 ـا1Blnk

Table 5-10: Classification results for Naskh font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	83	83	0	100.0	0.0	0	0	100.0	100.0	
آ	10	10	0	100.0	0.0	0	0	100.0	100.0	
أ	478	478	0	100.0	0.0	0	0	100.0	100.0	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	42	1	97.7	2.3	0	0	97.7	97.7	1ـت
ا	2091	2085	6	99.7	0.3	26	2	98.5	98.4	1ـء 1ـة 2ـم 2ـه 26ـحذف
ب	430	396	34	92.1	7.9	12	4	89.3	88.4	2ـت 1ـخ 1ـل 1ـم 2ـن 27ـي 12ـحذف
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	419	415	4	99.0	1.0	1	0	98.8	98.8	4ـن 1ـحذف
ث	122	118	4	96.7	3.3	2	0	95.1	95.1	4ـت 2ـحذف
ج	170	164	6	96.5	3.5	0	0	96.5	96.5	5ـح 1ـع
ح	234	205	29	87.6	12.4	0	0	87.6	87.6	7ـت 2ـج 11ـخ 1ـص 3ـع 2ـف 2ـم 1ـن
خ	113	102	11	90.3	9.7	0	1	90.3	89.4	1ـت 5ـح 5ـق
د	344	344	0	100.0	0.0	0	0	100.0	100.0	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	692	10	98.6	1.4	0	0	98.6	98.6	6ـد 4ـو
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	638	635	3	99.5	0.5	2	0	99.2	99.2	2ـا 1ـر 2ـحذف
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	415	410	5	98.8	1.2	0	0	98.8	98.8	1ـج 3ـض 1ـع
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	66	2	97.1	2.9	0	0	97.1	97.1	1ـظ 1ـل
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	817	807	10	98.8	1.2	1	0	98.7	98.7	4ـح 1ـخ 2ـص 2ـم 1ـه 1ـحذف
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	494	493	1	99.8	0.2	1	0	99.6	99.6	1ـق 1ـحذف
ق	465	463	2	99.6	0.4	2	0	99.1	99.1	2ـن 2ـحذف
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2104	2098	6	99.7	0.3	32	3	98.2	98.1	2ـب 1ـط 3ـم 32ـحذف
م	945	910	35	96.3	3.7	60	9	90.0	89.0	9ـا 6ـب 6ـة 3ـت 1ـغ 1ـف 7ـل 2ـن 60ـحذف
ن	1006	990	16	98.4	1.6	17	2	96.7	96.5	1ـب 10ـت 1ـث 4ـم 17ـحذف
ه	665	663	2	99.7	0.3	0	0	99.7	99.7	2ـم
و	937	936	1	99.9	0.1	0	0	99.9	99.9	1ـر
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	39	1	97.5	2.5	0	0	97.5	97.5	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	205	205	0	100.0	0.0	2	0	99.0	99.0	2ـحذف
ى	413	411	2	99.5	0.5	0	0	99.5	99.5	1ـر 1ـل
ي	1144	1133	11	99.0	1.0	15	0	97.7	97.7	3ـب 3ـم 1ـن 1ـو 3ـى 15ـحذف
Blnk	4609	4608	1	100.0	0.0	28	0	99.4	99.4	1ـل 28ـحذف
ﷲ	491	490	1	99.8	0.2	0	0	99.8	99.8	1ـه
Ins			21							2ـا 4ـب 1ـخ 3ـل 9ـم 2ـن

5.9.1.5 Classification of Simplified Font Text

The classification results for the Simplified Arabic font are shown in Table 5-11. The correctness of the Simplified Arabic font reached 99.82 and the accuracy reached 99.70. It could be considered as one of the best fonts in terms of the recognition rates. Three letters had some errors plus some insertions. Two of the letters have been substituted by a letter that has the same basic shape (once each). The letter Alef-Maqsoura ى has been replaced six times by the letter Yaa ي which has the same basic shape except for extra dots beneath the letter. The six replacements were out of 413 cases. There were 32 insertions, 21 of which were blank insertions. The blank problem is common for HMM based techniques. However, this is much more compensated for, by the major benefit of HMM technique which does not require segmentation of text and which can handle even touching characters.

5.9.1.6 Classification of Traditional Arabic Font Text

Table 5-12 shows the results of the classification for the Traditional Arabic font. The correctness percentage is 98.87 and the accuracy percentage is 98.83 for this font. As has been mentioned earlier (see Section 5.9), using one code for the different shapes of a character after recognition improves the recognition rate of single fonts. This font (Traditional Arabic) has the lowest improvement of the recognition rate compared with all other used fonts. Actually the effect is minimal. Twenty two letters had errors plus ten insertions and 117 deletions where half of them were for the letters *Meem* م and *Lam* ل. Most of the letters that have been substituted were substituted by letters that have the same basic shape.

Table 5-11: Classification results for Simplified Arabic font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	83	83	0	100.0	0.0	0	0	100.0	100.0	
آ	10	4	6	40.0	60.0	0	0	40.0	40.0	
أ	483	465	18	96.3	3.7	0	0	96.3	96.3	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	155	155	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2101	2100	1	100.0	0.0	0	3	100.0	99.8	-1Blnk
ب	409	409	0	100.0	0.0	0	0	100.0	100.0	
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	419	1	99.8	0.2	0	0	99.8	99.8	ـث1
ث	124	124	0	100.0	0.0	0	0	100.0	100.0	
ج	170	170	0	100.0	0.0	0	0	100.0	100.0	
ح	234	234	0	100.0	0.0	0	0	100.0	100.0	
خ	113	113	0	100.0	0.0	0	0	100.0	100.0	
د	344	344	0	100.0	0.0	0	1	100.0	99.7	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	702	0	100.0	0.0	0	0	100.0	100.0	
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	640	0	100.0	0.0	0	0	100.0	100.0	
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	415	415	0	100.0	0.0	0	0	100.0	100.0	
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	68	0	100.0	0.0	0	0	100.0	100.0	
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	818	0	100.0	0.0	0	0	100.0	100.0	
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	495	495	0	100.0	0.0	0	0	100.0	100.0	
ق	467	466	1	99.8	0.2	0	0	99.8	99.8	ـف1
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2134	2134	0	100.0	0.0	0	7	100.0	99.7	
م	1005	1005	0	100.0	0.0	0	0	100.0	100.0	
ن	1023	1022	1	99.9	0.1	0	0	99.9	99.9	ـل1
ه	663	663	0	100.0	0.0	0	0	100.0	100.0	
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	40	0	100.0	0.0	0	0	100.0	100.0	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	0	100.0	100.0	
ى	413	407	6	98.5	1.5	0	0	98.6	98.6	6ـي
ي	1157	1157	0	100.0	0.0	0	0	100.0	100.0	
Blnk	4637	4637	0	100.0	0.0	0	21	100.0	99.6	
ﷲ	491	490	1	99.8	0.2	0	0	99.8	99.8	ـه1
Ins			32							3ـا 1ـد 7ـل -21Blnk

Table 5-12: Classification results for Traditional Arabic font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	83	83	0	100.0	0.0	0	0	100.0	100.0	
آ	10	10	0	100.0	0.0	0	0	100.0	100.0	
أ	478	477	1	99.8	0.2	6	0	98.5	98.5	6ـحذف
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2096	2091	5	99.8	0.2	23	3	98.7	98.5	1ـت 3ـد 1ـف 23ـحذف
ب	429	405	24	94.4	5.6	8	0	92.5	92.5	1ـش 2ـل 2ـن 19ـي 8ـحذف
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	416	4	99.0	1.0	0	0	99.1	99.1	4ـن
ث	123	123	0	100.0	0.0	1	0	99.2	99.2	1ـحذف
ج	170	156	14	91.8	8.2	0	0	91.8	91.8	1ـث 11ـح 2ـخ
ح	234	197	37	84.2	15.8	0	0	84.2	84.2	3ـث 5ـج 28ـخ 1ـض
خ	113	111	2	98.2	1.8	0	1	98.2	97.4	2ـح
د	344	343	1	99.7	0.3	0	1	99.7	99.4	1ـذ
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	701	1	99.9	0.1	0	0	99.9	99.9	1ـز
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	638	2	99.7	0.3	0	0	99.7	99.7	2ـم
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	415	410	5	98.8	1.2	0	0	98.8	98.8	5ـض
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	67	67	0	100.0	0.0	0	0	100.0	100.0	
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	817	1	99.9	0.1	0	0	99.9	99.9	1ـم
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	495	495	0	100.0	0.0	0	0	100.0	100.0	
ق	467	467	0	100.0	0.0	0	0	100.0	100.0	
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2129	2127	2	99.9	0.1	6	1	99.6	99.6	1ـا 1ـم 6ـحذف
م	947	946	1	99.9	0.1	55	1	94.1	94.0	1ـن 55ـحذف
ن	1008	1004	4	99.6	0.4	15	1	98.1	98.0	1ـب 2ـل 1ـي 15ـحذف
ه	662	659	3	99.5	0.5	3	0	99.1	99.1	2ـم 1ـن 3ـحذف
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	40	0	100.0	0.0	0	0	100.0	100.0	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	0	100.0	100.0	
ى	413	407	6	98.5	1.5	0	0	98.6	98.6	3ـل 3ـي
ي	1147	1140	7	99.4	0.6	11	0	98.4	98.4	3ـن 4ـى 11ـحذف
Blnk	4634	4633	1	100.0	0.0	3	2	99.9	99.9	1ـا 3ـحذف
الله	491	490	1	99.8	0.2	0	0	99.8	99.8	1ـه
Ins			10							Blnk3ـا 1ـخ 1ـد 1ـل 1ـم 1ـن 2-

5.9.1.7 Classification of Tahoma Font Text

Table 5-13 shows the classification results for the Tahoma font, similar to the Arial font, Tahoma's correctness reached 99.92% and the accuracy reached 99.68%. Five letters resulted in some errors plus some insertions. The letter ا was substituted by the letter ث once out of 2113 instances. The letter ت was replaced by ث once out of 420 characters. Again, the only difference between the two letters is that the first letter has 2 dots above it and the second letter has three dots. The letter ج in this font has been substituted by the letter ح. Both letters have the same basic shape except for the dots in the body of the letter ج. The reverse substitution (i.e. ج was recognized as ح) has appeared 13 times. The letter ط has been substituted by the letter ر. The insertion of 46 instances of the letter ل in this font needs some explanation. The للّه ligature in Tahoma font resulted in the insertion of the letter *Lam* ل as the first two letters of the ligature are actually two consequent *Lams* as shown in Table 9. As the two letters are small and narrow, it recognized them as one lam and hence needed to insert another Lam.

Table 5-13: Classification results for Tahoma font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	83	83	0	100.0	0.0	0	1	100.0	98.8	
آ	10	10	0	100.0	0.0	0	0	100.0	100.0	
أ	484	484	0	100.0	0.0	0	0	100.0	100.0	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2113	2112	1	100.0	0.0	0	2	100.0	99.9	1ـث
ب	409	409	0	100.0	0.0	0	0	100.0	100.0	
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	419	1	99.8	0.2	0	0	99.8	99.8	1ـث
ث	123	123	0	100.0	0.0	0	0	100.0	100.0	
ج	170	169	1	99.4	0.6	0	0	99.4	99.4	1ـح
ح	234	221	13	94.4	5.6	0	0	94.4	94.4	13ـج
خ	113	113	0	100.0	0.0	0	0	100.0	100.0	
د	344	344	0	100.0	0.0	0	3	100.0	99.1	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	702	0	100.0	0.0	0	0	100.0	100.0	
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	640	0	100.0	0.0	0	0	100.0	100.0	
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	415	415	0	100.0	0.0	0	0	100.0	100.0	
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	67	1	98.5	1.5	0	1	98.5	97.1	1ـر
ظ	16	16	0	100.0	0.0	0	0	100.0	100.0	
ع	818	818	0	100.0	0.0	0	0	100.0	100.0	
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	495	495	0	100.0	0.0	0	0	100.0	100.0	
ق	467	467	0	100.0	0.0	0	0	100.0	100.0	
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2136	2136	0	100.0	0.0	0	46	100.0	97.9	
م	1005	1005	0	100.0	0.0	0	0	100.0	100.0	
ن	1016	1016	0	100.0	0.0	0	0	100.0	100.0	
ه	665	665	0	100.0	0.0	0	0	100.0	100.0	
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	40	0	100.0	0.0	0	0	100.0	100.0	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	0	100.0	100.0	
ى	413	413	0	100.0	0.0	0	0	100.0	100.0	
ي	1159	1159	0	100.0	0.0	0	0	100.0	100.0	
Blnk	4632	4632	0	100.0	0.0	0	1	100.0	100.0	
ﷲ	491	490	1	99.8	0.2	0	0	99.8	99.8	1ـه
Ins			54							Blnk1ـء 2ـا 3ـد 1ـط 46ـل 1-

5.9.1.8 Classification of Thuluth Font Text

Table 5-14 shows the classification results for the Thuluth font. The correctness for this font was 98.85% and the accuracy reached 98.78%. The effect of using one code for the different shapes of a character on improving the recognition rate is the second highest in this font compared to all the fonts used. The reason is due to the greater variation of character shapes in this font compared with others. As this font is tightly cursive and has a lot of overlapping, there were around 260 substitutions and 15 insertions. Investigation of the cases of the substitutions shows that most of the cases could be justified. The shapes of characters with common basic shapes that differ in only the number of dots used were the common characteristics for most of the errors (see Table 5-15 for characters with common basic shapes). Nevertheless, the accuracy is 98.78.

Table 5-14: Classification results for Thuluth font.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	77	77	0	100.0	0.0	0	2	100.0	97.4	
آ	10	10	0	100.0	0.0	0	0	100.0	100.0	
أ	484	483	1	99.8	0.2	0	0	99.8	99.8	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	42	42	0	100.0	0.0	0	0	100.0	100.0	
ا	2112	2111	1	100.0	0.0	0	0	100.0	100.0	1ـه
ب	428	402	26	93.9	6.1	0	0	93.9	93.9	2ـل 17ـن 7ـي
ة	230	230	0	100.0	0.0	0	0	100.0	100.0	
ت	417	408	9	97.8	2.2	0	1	97.8	97.6	1ـج 5ـل 3ـن
ث	123	123	0	100.0	0.0	0	0	100.0	100.0	
ج	170	165	5	97.1	2.9	0	0	97.1	97.1	4ـح 1ـخ
ح	233	191	42	82.0	18.0	0	0	82.0	82.0	30ـج 12ـخ
خ	113	111	2	98.2	1.8	0	0	98.2	98.2	1ـج 1ـح
د	344	341	3	99.1	0.9	0	1	99.1	98.8	3ـذ
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	666	36	94.9	5.1	0	0	94.9	94.9	1ـم 35ـن
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	639	1	99.8	0.2	0	0	99.8	99.8	1ـر
ش	119	118	1	99.2	0.8	0	0	99.2	99.2	1ـن
ص	413	412	1	99.8	0.2	0	0	99.8	99.8	1ـم
ض	93	92	1	98.9	1.1	0	0	98.9	98.9	1ـم
ط	68	66	2	97.1	2.9	0	0	97.1	97.1	1ـظ 1ـم
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	814	4	99.5	0.5	0	0	99.5	99.5	4ـخ
غ	44	43	1	97.7	2.3	0	0	97.7	97.7	1ـع
ف	495	495	0	100.0	0.0	0	4	100.0	99.2	
ق	467	467	0	100.0	0.0	0	0	100.0	100.0	
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2130	2125	5	99.8	0.2	0	2	99.8	99.7	2ـت 1ـم 1ـن 1ـي
م	968	951	17	98.2	1.8	0	1	98.2	98.1	1ـب 8ـح 2ـر 3ـن 2ـه 1ـي
ن	1013	1004	9	99.1	0.9	0	0	99.1	99.1	5ـت 2ـض 2ـل
ه	664	658	6	99.1	0.9	0	0	99.1	99.1	5ـم 1ـن
و	937	935	2	99.8	0.2	0	0	99.8	99.8	1ـر 1ـم
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	40	0	100.0	0.0	0	0	100.0	100.0	
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	0	100.0	100.0	
ى	413	368	45	89.1	10.9	0	0	89.1	89.1	1ـئ 5ـر 39ـم
ي	1156	1152	4	99.7	0.3	0	0	99.7	99.7	3ـب 1ـش
Blnk	4610	4577	33	99.3	0.7	0	4	99.3	99.2	3ـأ 2ـد 3ـذ 25ـر
ﷲ	491	490	1	99.8	0.2	0	0	99.8	99.8	1ـه
Ins			15							Blnk2ـء 1ـت 1ـد 4ـف 2ـل 1ـم 4-

5.9.1.9 Comparisons

Table 5-16 summarizes the recognition results for Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic font texts. The table shows the average correctness and accuracy for all these fonts. These are the averages of the recognition rates (correctness and accuracy) of the classifications of the nine testing and training sets for each font using the dataset PATS-A01. The average for each font for each run was computed as the average of all shapes under test.

Table 5-15: Arabic characters with common basic shapes in most fonts.

Basic shape	Characters
ا	ا ا إ ـلـ
ب	ب ت ث ـنـ ـيـ
ح	ج ح خ
د	د ذ
ر	ر ز
س	س ش
ص	ص ض
ط	ط ظ
ع	ع غ
ف	ف ق
ك	ك ل
ه	ه ء ـمـ
ى	ى ي
لا	لا لأ لآ لإ

Table 5-16: Results for Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic fonts for dataset PATS-A01 of all training & testing sets.

	Arial		Tahoma		Akhbar		Thuluth		Naskh		Simplified Arabic		Traditional Arabic		Andalus	
Let	Corr.	Acc.	Corr.	Acc.	Corr.	Acc.	Corr.	Acc.	Corr.	Acc.	Corr.	Acc.	Corr.	Acc.	Corr.	Acc.
ء	100	100	100	98.8	96.3	96.3	100	97.4	100	100	100	100	100	100	100	100
آ	80	80	100	100	60	60	100	100	100	100	40	40	100	100	100	100
أ	100	100	100	100	99.6	99.6	99.8	99.8	100	100	96.3	96.3	98.5	98.5	100	100
ؤ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
إ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ئ	100	100	100	100	100	100	100	100	97.7	97.7	100	100	100	100	100	100
ا	100	100	100	99.9	99.1	99	100	100	98.5	98.4	100	99.8	98.7	98.5	100	100
ب	100	100	100	100	99.3	99.3	93.9	93.9	89.3	88.4	100	100	92.5	92.5	100	99.8
ة	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ت	100	100	99.8	99.8	100	100	97.8	97.6	98.8	98.8	99.8	99.8	99.1	99.1	100	100
ث	100	100	100	100	98.4	92.6	100	100	95.1	95.1	100	100	99.2	99.2	100	100
ج	100	100	99.4	99.4	100	100	97.1	97.1	96.5	96.5	100	100	91.8	91.8	100	100
ح	98.3	98.3	94.4	94.4	100	100	82	82	87.6	87.6	100	100	84.2	84.2	100	100
خ	100	100	100	100	100	100	98.2	98.2	90.3	89.4	100	100	98.2	97.4	100	100
د	100	99.7	100	99.1	100	99.7	99.1	98.8	100	100	100	99.7	99.7	99.4	100	99.7
ذ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ر	100	100	100	100	100	100	94.9	94.9	98.6	98.6	100	100	99.9	99.9	100	100
ز	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
س	100	100	100	100	99.8	98.9	99.8	99.8	99.2	99.2	100	100	99.7	99.7	100	100
ش	100	100	100	100	99.2	99.2	99.2	99.2	100	100	100	100	100	100	100	100
ص	100	100	100	100	100	100	99.8	99.8	98.8	98.8	100	100	98.8	98.8	100	100
ض	100	100	100	100	100	100	98.9	98.9	100	100	100	100	100	100	100	100
ط	100	100	98.5	97.1	100	100	97.1	97.1	97.1	97.1	100	100	100	100	100	100
ظ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ع	100	100	100	100	99.8	99.8	99.5	99.5	98.7	98.7	100	100	99.9	99.9	100	100
غ	100	100	100	100	97.7	97.7	97.7	97.7	100	100	100	100	100	100	100	100
ف	100	100	100	100	99.6	99.6	100	99.2	99.6	99.6	100	100	100	100	100	100
ق	100	100	100	100	98.9	98.9	100	100	99.1	99.1	99.8	99.8	100	100	100	100
ك	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ل	100	99.9	100	97.9	98.1	97.9	99.8	99.7	98.2	98.1	100	99.7	99.6	99.6	100	77.7
م	100	100	100	100	99.9	99.9	98.2	98.1	90	89	100	100	94.1	94	100	100
ن	100	100	100	100	99.4	99.3	99.1	99.1	96.7	96.5	99.9	99.9	98.1	98	100	100
ه	99.7	99.7	100	100	99.7	99.7	99.1	99.1	99.7	99.7	100	100	99.1	99.1	100	100
و	100	100	100	100	100	100	99.8	99.8	99.9	99.9	100	100	100	100	100	100
لآ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
لأ	90	90	100	100	100	100	100	100	97.5	97.5	100	100	100	100	100	100
لإ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
لا	100	98.1	100	100	100	100	100	100	99	99	100	100	100	100	100	100
ى	100	100	100	100	100	100	89.1	89.1	99.5	99.5	98.6	98.6	98.6	98.6	100	100
ي	100	100	100	100	98.4	98.4	99.7	99.7	97.7	97.7	100	100	98.4	98.4	100	100
B	100	100	100	100	100	100	99.3	99.2	99.4	99.4	100	99.6	99.9	99.9	99.9	99.9
ﷲ	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	100	100
T	**99.9**	**99.9**	**99.9**	**99.7**	**99.4**	**99.3**	**98.9**	**98.8**	**98.2**	**98.1**	**99.8**	**99.7**	**98.9**	**98.8**	**100**	**97.9**

5.10 Summary and Conclusions

This chapter presents the result of automatic recognition of off-line Arabic text recognition based on estimating simple and effective features that are compatible with HMM-based OCR. The chapter includes performance analyses using the HMM with different numbers of features, different sizes of sliding windows, different numbers of states and different codebook sizes. We applied this technique to each of the eight Arabic fonts under study.

Two database sets of line images were used for testing and training. The first one consists of 2766 line images where 2500 line images were used for training and the remaining 266 for testing. The second database set consists of 318 line images where 286 line images were used for testing and the remaining 32 line images were used for training. The test line images were randomly selected. The remaining unselected line images were assigned for training. Nine testing and training sets were used. This chapter reported the results obtained using the first dataset for single fonts.

The experimental results, discussed earlier (see Section 5.9.1), indicated the effectiveness of the proposed technique in the automatic recognition of off-line printed Arabic text with different types of fonts. They show the effectiveness of our features. We used a small number of simple and effective features that can be computed quickly. This was repeated for all vertical strips with an overlap of one pixel. Ten, sixteen, and thirty features were extracted in different experiments from each vertical strip of the text line image. For single fonts the three schemes (ten, sixteen, and thirty) were suitable.

We applied our technique to eight different Arabic fonts. They all gave acceptable recognition rates. For single font recognition, the accuracy percentages were: 99.9 for Arial, 99.68 for Tahoma, 99.34 for Akhbar, 98.78 for Thuluth, 98.09 for Naskh, 99.7 for Simplified Arabic, 98.83 for Traditional Arabic, and 97.86 for Andalus. We believe, and up to the author's knowledge, these results are new records in the recognition of printed Arabic text.

Several aspects of our technique resulted in the high recognition rates. Our technique is based on a novel hierarchical sliding window technique with overlapping and non-overlapping windows. We considered each shape of an Arabic character as a separate class, not combining multiple shapes in one class as done by other researchers. The number of classes became 126 compared with 40 classes if all the shapes of a character are considered as separate classes. Some basic ligatures were also included. This technique does not require the segmentation of Arabic cursive text which is known to be problematic as errors in segmentation could increase the errors in recognitions. Hence, using this technique, segmentation was a by-product of our technique. Finally, the presented technique is language independent as we are going to demonstrate in the next chapter.

The next chapter reports the classification results of multi-font recognition using the same methodology we have presented in this chapter. It also reports the classification of English and Bangla languages using the same proposed methodology to show that our proposed feature extraction schemes are language independent.

Chapter 6. Multi-font recognition and Work with other Languages

6.1 Introduction

The classifications of multi-fonts are discussed in this chapter. Then the chapter presents the classifications of English and Bangla using the proposed techniques for Arabic OCR. This chapter is structured as follows. Section 6.2 discusses multi-font classifications. Section6.3 introduces the work with other languages. The English dataset used is described in Section 6.4. Section 6.5 describes the Bangla datasets. Section 6.6 presents and discusses the classification results. Section 6.7 presents the summary of the chapter.

6.2 Multi-font Classification

The extension of a single font feature set and model to multi-font is addressed in this section. Analysis of common attributes between multi-font and single font has been conducted. Based on the results of the analysis it has been noticed that there is a need to categorise the fonts as families and experiment on each family alone with the same set of features. Some font styles look totally different from other font styles. As the developed set of features is based mainly on the density distribution of the pixels of the text, some differences are expected. Categorizing fonts of similar styles increased the recognition rates. Moreover, to the author’s knowledge, this is an area of research that was not addressed by other researchers and no published work/results currently exist.

The experiments for multi-font training and testing were pursued on the PATS-A02 dataset (see Section 3.5). The thirty feature scheme was used (see Section 4.4) for feature extraction. Nine training and testing sets were prepared for each multi-font category. For each font, 32 line images were selected randomly for testing. The remaining 286 line images of the dataset were assigned for training. The training set for a given multi-font category consisted of all the training sets of all the fonts in this category. The testing set for the category consisted of all the testing sets of all fonts in the category. Each training and testing set of the nine sets of all-fonts category (8 fonts) consisted of 256 line images for testing (8 x 32) and 2288 line images for training (8 x 286). Each training and testing set of the nine sets of a multi-font category of any three fonts consisted of 96 line images for testing (3 x 32) and 858 line images for training (3 x 286).

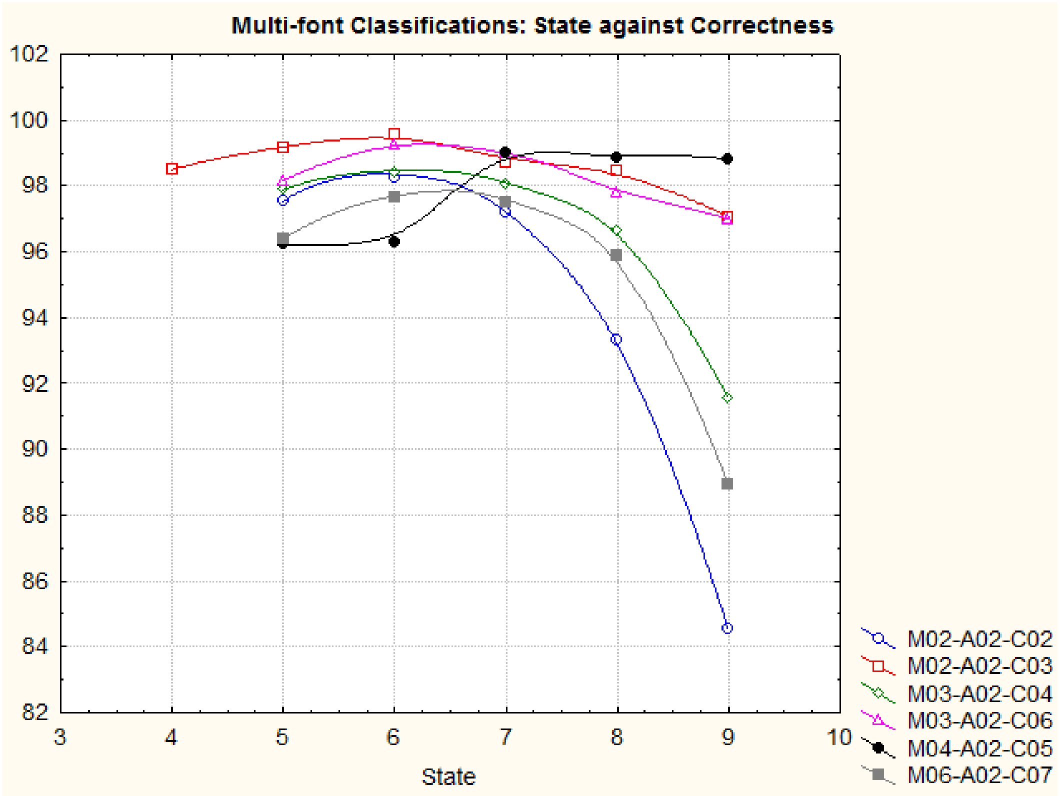


Figure 6.1: States versus correctness for 7 multi-font categories.

Taking into account the characteristics of each font, several font combinations have been experimented with. The most promising categories with respect to the recognition rates are shown in Table 6-1. Figure 6.1 shows the correctness percentage of these categories for different numbers of states. The following subsections discuss the classifications of these seven categories.

Table 6-1: Multi-font categories.

Category	Fonts
M08-A02-C01	Akhbar, Andalus, Simplified, Traditional, Arial, Tahoma, Naskh, & Thuluth
M02-A02-C02	Naskh & Thuluth
M02-A02-C03	Arial & Tahoma
M03-A02-C04	Arial, Tahoma, & Traditional
M04-A02-C05	Akhbar, Andalus, Simplified, & Traditional
M03-A02-C06	Akhbar, Andalus, & Simplified
M06-A02-C07	Akhbar, Andalus, Simplified, Traditional, Arial, and Tahoma

6.2.1 All 8 fonts Classification (M08-A02-C01)

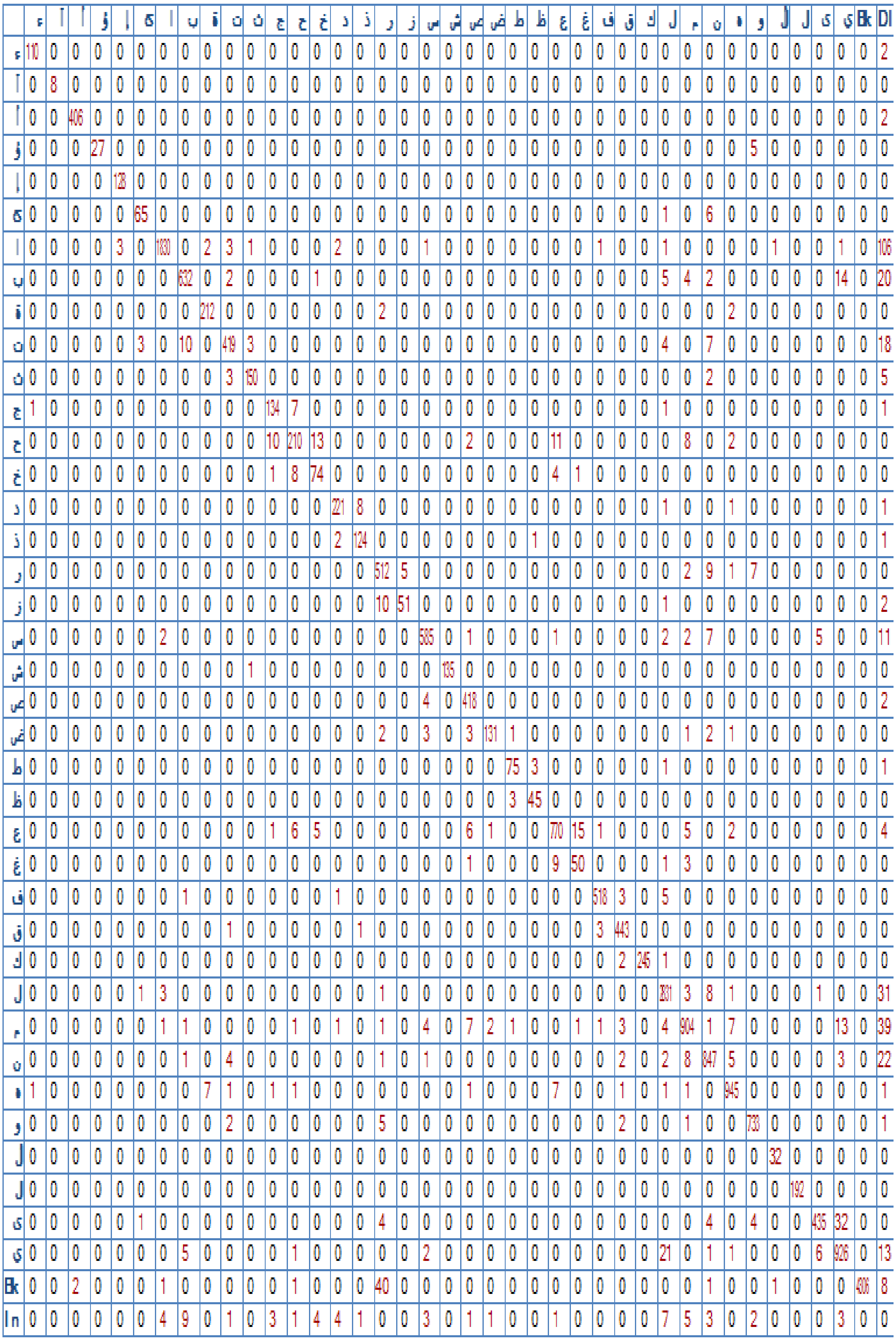
Table 6-2 shows the best combinations of codebook size and the number of HMM states for the eight fonts (M-08-A02-C01) obtained experimentally with the correctness and accuracy percentages. Large numbers of experiments were carried out. Besides combining the number of states and codebook sizes, different line image normalization heights were also considered including 80 pixels, 120 pixels, 150 pixels, and un-normalized heights. Moreover, extensive experiments on the used feature extraction scheme were carried out. It is worth pointing out that these results were obtained using the thirty feature extraction scheme (see Section 4.4).

As the raw confusion matrix for the shapes cannot be physically displayed, the confusion matrix for the letters after collapsing their shapes into one code is shown in Table 6-3. This matrix is shown as a sample for the hundreds of similar resulting matrices. We choose this table as a sample because it is the richest matrix with respect to errors. Raw confusion matrices and detailed analysis for each run for each line image are provided in the enclosed CD-ROM (See Appendix A). Table 6-4 shows the summary of the raw confusion matrix of Table 6-3 in more informative way. It shows for each letter (after collapsing its shapes) the number of samples used in testing, the correctly recognized samples, the wrongly recognized, the wrongly deleted, the wrongly inserted, the correctness and accuracy percentages and the letters that have been wrongly recognized.

Table 6-2: Classification/recognition information for M08-A02-C01.

Codebook	States	Correctness	Accuracy
224	5	93.32	92.12
224	6	95.63	95.03
224	**7**	**95.85**	**95.61**
224	8	93.21	92.93
224	9	85.1	84.87

Table 6-3: The confusion matrix for the multi-font recognition of the 8 fonts (M08-A02-C01)



	ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	لٓ	ل	ى	ي	Bk	Dl
ء	110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
آ	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
أ	0	0	406	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
ؤ	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
إ	0	0	0	0	128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ئ	0	0	0	0	0	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	6	0	0	0	0	0	0	0	0
ا	0	0	0	0	3	0	1830	0	2	3	1	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	106
ب	0	0	0	0	0	0	0	632	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	2	0	0	0	0	0	14	0	20
ة	0	0	0	0	0	0	0	0	212	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
ت	0	0	0	0	0	3	0	10	0	419	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	7	0	0	0	0	0	0	0	18
ث	0	0	0	0	0	0	0	0	0	3	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	5
ج	1	0	0	0	0	0	0	0	0	0	0	134	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
ح	0	0	0	0	0	0	0	0	0	0	0	10	210	13	0	0	0	0	0	0	2	0	0	0	11	0	0	0	0	0	8	0	2	0	0	0	0	0	0	0
خ	0	0	0	0	0	0	0	0	0	0	0	1	8	74	0	0	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
د	0	0	0	0	0	0	0	0	0	0	0	0	0	0	221	8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1
ذ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	124	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ر	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	512	5	0	0	0	0	0	0	0	0	0	0	0	0	2	9	1	7	0	0	0	0	0	0
ز	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	51	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2
س	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	585	0	1	0	0	0	1	0	0	0	0	2	2	7	0	0	0	0	5	0	0	11
ش	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	135	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ص	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	418	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
ض	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3	0	3	131	1	0	0	0	0	0	0	0	1	2	1	0	0	0	0	0	0	0
ط	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
ظ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ع	0	0	0	0	0	0	0	0	0	0	0	1	6	5	0	0	0	0	0	0	6	1	0	0	770	15	1	0	0	0	5	0	2	0	0	0	0	0	0	4
غ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	9	50	0	0	0	1	3	0	0	0	0	0	0	0	0	0
ف	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	518	3	0	5	0	0	0	0	0	0	0	0	0	0
ق	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	443	0	0	0	0	0	0	0	0	0	0	0	0
ك	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	245	1	0	0	0	0	0	0	0	0	0	0
ل	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2831	3	8	1	0	0	0	1	0	0	31
م	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1	0	4	0	7	2	1	0	0	1	1	3	0	4	904	1	7	0	0	0	0	13	0	39
ن	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	2	0	2	8	847	5	0	0	0	0	3	0	22
ه	1	0	0	0	0	0	0	0	7	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	7	0	0	1	0	1	1	0	945	0	0	0	0	0	0	1
و	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	733	0	0	0	0	0	1
لٓ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0
ل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	192	0	0	0	0
ى	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	0	0	435	32	0	0
ي	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	21	0	1	1	0	0	0	6	926	0	13
Bk	0	0	2	0	0	0	1	0	0	0	0	0	1	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	406	8
In	0	0	0	0	0	0	4	9	0	1	0	3	1	4	4	1	0	0	3	0	1	1	0	0	1	0	0	0	0	7	5	3	0	2	0	0	0	3	0	0

Table 6-4: Classification results for M08-A02-C01 multi-font category (8 fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	110	110	0	100.00	0.00	2	0	98.18	98.18	-2Del+
آ	8	8	0	100.00	0.00	0	0	100	100	
أ	406	406	0	100.00	0.00	2	0	99.51	99.51	-2Del+
ؤ	32	27	5	84.38	15.63	0	0	84.38	84.38	+ـو5
إ	128	128	0	100.00	0.00	0	0	100	100	
ئ	72	65	7	90.28	9.72	0	0	90.28	90.28	+ـن6 +ـل1
ا	1846	1830	16	99.13	0.87	106	4	93.39	93.17	+ـلأ1 +ـل1 +ـف1 +ـس1 +ـد2 +ـث1+ـت3+ـة2 -106 +ـي1Del+
ب	660	632	28	95.76	4.24	20	9	92.73	91.36	-20 +ـي14 +ـن2+ـم4+ـل5 +ـخ1 +ـت2Del+
ة	216	212	4	98.15	1.85	0	0	98.15	98.15	+ـه2 +ـر2
ت	446	419	27	93.95	6.05	18	1	89.91	89.69	-18 +ـن7 +ـل4 +ـث3 +ـب10 +ـئ3Del+
ث	155	150	5	96.77	3.23	5	0	93.55	93.55	-5 +ـن2 +ـت3Del+
ج	143	134	9	93.71	6.29	1	3	93.01	90.91	-1 +ـل1 +ـح7 +ـء1 Del+
ح	256	210	46	82.03	17.97	0	1	82.03	81.64	+ـه2 +ـم8 +ـع11 +ـص2 +ـخ13 +ـج10
خ	88	74	14	84.09	15.91	0	4	84.09	79.55	+ـغ1+ـع4 +ـح8+ـج1
د	231	221	10	95.67	4.33	1	4	95.24	93.51	-1 +ـه1 +ـل1 +ـذ8Del+
ذ	127	124	3	97.64	2.36	1	1	96.85	96.06	-1 +ـظ1 +ـد2Del+
ر	536	512	24	95.52	4.48	0	0	95.52	95.52	+ـو7+ـه1+ـن9+ـم2 +ـز5
ز	62	51	11	82.26	17.74	2	0	79.03	79.03	-2 +ـل1 +ـر10Del+
س	605	585	20	96.69	3.31	11	3	94.88	94.38	+ـى5 +ـن7+ـم2+ـل2 +ـع1 +ـص1 +ـا2 -11Del+
ش	136	135	1	99.26	0.74	0	0	99.26	99.26	+ـث1
ص	422	418	4	99.05	0.95	2	1	98.58	98.34	-2 +ـس4Del+
ض	144	131	13	90.97	9.03	0	1	90.97	90.28	+ـه1+ـن2+ـم1 +ـط1 +ـص3 +ـس3 +ـر2
ط	79	75	4	94.94	5.06	1	0	93.67	93.67	-1 +ـل1 +ـظ3Del+
ظ	48	45	3	93.75	6.25	0	0	93.75	93.75	+ـط3
ع	812	770	42	94.83	5.17	4	1	94.33	94.21	+ـف1+ـغ15 +ـض1+ـص6 +ـخ5+ـح6+ـج1 -4 +ـه2 +ـم5Del+
غ	64	50	14	78.13	21.88	0	0	78.13	78.13	+ـم3+ـل1 +ـع9 +ـص1
ف	528	518	10	98.11	1.89	0	0	98.11	98.11	+ـل5 +ـق3 +ـد1 +ـب1
ق	448	443	5	98.88	1.12	0	0	98.88	98.88	+ـف3 +ـذ1 +ـت1
ك	248	245	3	98.79	1.21	0	0	98.79	98.79	+ـل1 +ـق2
ل	2849	2831	18	99.37	0.63	31	7	98.28	98.03	-31 +ـى1 +ـه1+ـن8+ـم3 +ـر1 +ـا3+ـئ1Del+
م	953	904	49	94.86	5.14	39	5	90.77	90.24	+ـس4 +ـر1 +ـد1 +ـح1 +ـب1+ـا1 +ـل4 +ـق3+ـف1+ـغ1 +ـط1+ـض2+ـص7 -39 +ـي13 +ـه7+ـن1Del+
ن	874	847	27	96.91	3.09	22	3	94.39	94.05	+ـه5 +ـم8+ـل2 +ـق2 +ـس1 +ـر1 +ـت4 +ـب1 -22 +ـي3Del+
ه	967	945	22	97.72	2.28	1	0	97.62	97.62	+ـق1 +ـع7 +ـص1 +ـح1+ـج1 +ـت1+ـة7 +ـء1 -1 +ـم1+ـل1Del+
و	743	733	10	98.65	1.35	1	2	98.52	98.25	-1 +ـم1 +ـق2 +ـر5 +ـت2Del+
لأ	32	32	0	100.00	0.00	0	0	100	100	
لا	192	192	0	100.00	0.00	0	3	100	98.44	
ى	480	435	45	90.63	9.38	0	0	90.63	90.63	+ـي32 +ـو4 +ـن4 +ـر4 +ـئ1
ي	963	926	37	96.16	3.84	13		94.81	94.81	+ـى6 +ـه1+ـن1 +ـل21 +ـس2 +ـح1 +ـب5 -13Del+
Blank	4352	4306	46	98.94	1.06	8	0	98.76	98.76	-8 +ـلأ1 +ـن1 +ـر40 +ـح1 +ـا1 +ـأ2Del+
Ins	53	0	53							+ـذ1+ـد4+ـخ4+ـح1+ـج3 +ـت1 +ـب9+ـا4 +ـو2 +ـن3+ـم5+ـل7 +ـع1 +ـض1+ـص1 +ـس3 +ـلا3

6.2.2 Classifications of Naskh and Thuluth (M02-A02-C02)

Naskh and Thuluth fonts have a lot of variation compared with other fonts. Reaching a 98% recognition rate for these two fonts is a new achievement. Table 6-5 shows the correctness and the accuracy percentages for the best cases we could reach with codebook size of 128 and 6 HMM states. Table 6-6 shows the analysis per letter for the two fonts. By studying this table, it can be seen that having the two fonts adds more confusion to the recognition process for some letters and less confusion for others. For example the letter خ has several misrecognition instances when each font is considered alone. However, when both fonts are considered, all instances of the letter have been recognized correctly. The same is true for the letters س, ش, ص, and ض. On the other hand, the letter ت has more misrecognition instances when multi-fonts are considered.

Table 6-5: Classification/recognition information for M02-A02-C02.

Codebook	States	Correctness	Accuracy
128	5	97.55	96.86
128	**6**	**98.27**	**98.12**
128	7	97.17	97.08
128	8	93.32	93.18
128	9	84.52	84.06

Table 6-6: Classification results for M02-A02-C02 multi-font category (2 Fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
آ	2	2	0	100.00	0.00	0	0	100	100	
أ	102	102	0	100.00	0.00	0	0	100	100	
ؤ	8	8	0	100.00	0.00	0	0	100	100	
إ	32	32	0	100.00	0.00	0	0	100	100	
ئ	18	17	1	94.44	5.56	0	0	94.44	94.12	+ـل1
ا	488	487	1	99.80	0.20	0	0	99.8	99.79	+ـس1
ب	170	158	12	92.94	7.06	3	0	91.18	90.51	-3 +ـي5+ـم2+ـل2 Del+
ة	54	54	0	100.00	0.00	0	1	100	100	
ت	116	108	8	93.10	6.90	1	0	92.24	91.67	+ـي1+ـل2+ـح1 +ـث2+ـئ1 -1Del+
ث	40	37	3	92.50	7.50	1	0	90	89.19	-1 +ـت2 Del+
ج	36	36	0	100.00	0.00	0	1	100	100	
ح	64	58	6	90.63	9.38	0	0	90.63	89.66	+ـن1+ـع2+ـخ1 +ـج2
خ	22	22	0	100.00	0.00	0	0	100	100	
د	58	56	2	96.55	3.45	0	0	96.55	96.43	+ـذ2
ذ	32	31	1	96.88	3.13	1	0	93.75	93.55	-1Del+
ر	134	134	0	100.00	0.00	0	0	100	100	
ز	16	16	0	100.00	0.00	0	0	100	100	
س	154	152	2	98.70	1.30	0	0	98.7	98.68	+ـح2
ش	34	34	0	100.00	0.00	0	0	100	100	
ص	106	106	0	100.00	0.00	0	0	100	100	
ض	36	36	0	100.00	0.00	0	0	100	100	
ط	20	19	1	95.00	5.00	0	0	95	94.74	+ـن1
ظ	12	12	0	100.00	0.00	0	1	100	100	
ع	204	204	0	100.00	0.00	0	0	100	100	
غ	16	16	0	100.00	0.00	0	0	100	100	
ف	132	128	4	96.97	3.03	0	0	96.97	96.88	+ـن1+ـم1+ـق1+ـب1
ق	112	111	1	99.11	0.89	0	0	99.11	99.1	+ـف1
ك	62	62	0	100.00	0.00	0	0	100	100	
ل	720	714	6	99.17	0.83	0	1	99.17	99.16	+ـم2+ـب4
م	248	243	5	97.98	2.02	2	2	97.18	97.12	-2+ـه1+ـل1+ـت1 Del+
ن	224	214	10	95.54	4.46	3	15	94.2	93.93	+ـي1+ـم1+ـل2 +ـق1 +ـت2 -3Del+
ه	242	240	2	99.17	0.83	0	0	99.17	99.17	+ـم1+ـس1
و	186	183	3	98.39	1.61	0	0	98.39	98.36	+ـر3
لأ	8	8	0	100.00	0.00	0	0	100	100	
لا	48	48	0	100.00	0.00	0	0	100	100	
ى	120	116	4	96.67	3.33	0	0	96.67	96.55	+ـر4
ي	244	243	1	99.59	0.41	0	0	99.59	99.59	+ـت1
	1086	1084	2	99.82	0.18	0	8	99.82	99.82	+ـر1 +ـأ1
Ins										+ـظ1+ـج1+ـة1 + -8+ـن15+ـم2+ـل1

6.2.3 Classifications of Arial and Tahoma (M02-A02-C03)

Table 6-7 shows the percentages of correctness and accuracy of the two fonts Arial and Tahoma. The size of the code book was 112 for these results and the best recognition rate was when 6 HMM states were used. Classification results of Arial and Tahoma fonts are comparable with their classification results for single fonts. The analysis of each letter for these two fonts is shown in Table 6-8.

Table 6-7: Classification/recognition information for M02-A02-C03.

Codebook	States	Correctness	Accuracy
112	4	98.49	96.56
112	5	99.14	98.75
112	**6**	**99.56**	**99.21**
112	7	98.71	98.53
112	8	98.44	98.29
112	9	97.06	96.91

Table 6-8: Classification results for M02-A02-C03 multi-font Category (2 Fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
آ	2	2	0	100.00	0.00	0	0	100	100	
أ	102	102	0	100.00	0.00	0	0	100	100	
ؤ	8	8	0	100.00	0.00	0	0	100	100	
إ	32	32	0	100.00	0.00	0	0	100	100	
ئ	18	18	0	100.00	0.00	0	0	100	100	
ا	488	488	0	100.00	0.00	0	7	100	100	
ب	170	169	1	99.41	0.59	0	0	99.41	99.41	+ـي1
ة	54	54	0	100.00	0.00	0	0	100	100	
ت	116	113	3	97.41	2.59	0	0	97.41	97.35	+ـن1+ـث2
ث	40	38	2	95.00	5.00	0	0	95	94.74	+ـت2
ج	36	36	0	100.00	0.00	0	0	100	100	
ح	64	63	1	98.44	1.56	0	0	98.44	98.41	+ـج1
خ	22	22	0	100.00	0.00	0	0	100	100	
د	58	58	0	100.00	0.00	0	0	100	100	
ذ	32	32	0	100.00	0.00	0	0	100	100	
ر	134	130	4	97.01	2.99	0	0	97.01	96.92	+ـن4
ز	16	16	0	100.00	0.00	0	0	100	100	
س	154	153	1	99.35	0.65	0	0	99.35	99.35	+ـص1
ش	34	34	0	100.00	0.00	0	0	100	100	
ص	106	106	0	100.00	0.00	0	0	100	100	
ض	36	36	0	100.00	0.00	0	0	100	100	
ط	20	20	0	100.00	0.00	0	0	100	100	
ظ	12	11	1	91.67	8.33	0	0	91.67	90.91	+ـط1
ع	204	203	1	99.51	0.49	0	0	99.51	99.51	+ـغ1
غ	16	16	0	100.00	0.00	0	0	100	100	
ف	132	132	0	100.00	0.00	0	0	100	100	
ق	112	112	0	100.00	0.00	0	0	100	100	
ك	62	62	0	100.00	0.00	0	0	100	100	
ل	720	719	1	99.86	0.14	0	1	99.86	99.86	+ـم1
م	248	245	3	98.79	1.21	2	2	97.98	97.96	-2+ـن1Del+
ن	224	224	0	100.00	0.00	0	0	100	100	
ه	242	242	0	100.00	0.00	0	0	100	100	
و	186	186	0	100.00	0.00	0	0	100	100	
لأ	8	8	0	100.00	0.00	0	0	100	100	
لا	48	48	0	100.00	0.00	0	0	100	100	
ى	120	120	0	100.00	0.00	0	0	100	100	
ي	244	240	4	98.36	1.64	0	0	98.36	98.33	+ـى4
	1090	1090	0	100.00	0.00	0	8	100	100	
Ins										+ -8+ـم2+ـل1+ـا7

6.2.4 Classifications of Arial, Tahoma and Traditional (M03-A02-C04)

The best recognition rate for this multi-font category (Arial, Tahoma, and Traditional) is achieved using 6 HMM states and a codebook of size 224. Table 6-9 shows the combinations for the best recognition rates for the three fonts. Table 6-10 shows the classification results per letter for this category. The author is not aware of any research publication that has reported a similar or better recognition rate.

Table 6-9: Classification/recognition information for M03-A02-C04.

Codebook	States	Correctness	Accuracy
224	5	97.89	97.61
224	**6**	**98.42**	**98.11**
224	7	98.04	97.85
224	8	96.65	96.59
224	9	91.54	91.37

Table 6-10: Classification results for M03-A02-C04 multi-font category (3 fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
آ	3	3	0	100.00	0.00	0	0	100	100	
أ	153	153	0	100.00	0.00	0	0	100	100	
ؤ	12	12	0	100.00	0.00	0	0	100	100	
إ	48	46	2	95.83	4.17	0	0	95.83	95.65	
ئ	27	27	0	100.00	0.00	0	0	100	100	
ا	732	716	16	97.81	2.19	15	10	95.77	95.67	-15 +ـت1 Del+
ب	255	251	4	98.43	1.57	1	3	98.04	98.01	-1 +ـي1+ـل1+ـا1Del+
ة	81	81	0	100.00	0.00	0	0	100	100	
ت	174	168	6	96.55	3.45	1	0	95.98	95.83	-1+ـن2 +ـل1+ـث2Del+
ث	60	59	1	98.33	1.67	0	0	98.33	98.31	+ـن1
ج	54	53	1	98.15	1.85	0	0	98.15	98.11	+ـح1
ح	96	85	11	88.54	11.46	0	1	88.54	87.06	+ـل1+ـخ4 +ـج6
خ	33	31	2	93.94	6.06	0	0	93.94	93.55	+ـح1+ـج1
د	87	85	2	97.70	2.30	0	0	97.7	97.65	+ -1+ـذ1
ذ	48	48	0	100.00	0.00	0	1	100	100	
ر	201	192	9	95.52	4.48	0	0	95.52	95.31	+ـن9
ز	24	21	3	87.50	12.50	0	0	87.5	85.71	+ـر3
س	231	227	4	98.27	1.73	0	1	98.27	98.24	+ـن3+ـع1
ش	51	51	0	100.00	0.00	0	1	100	100	
ص	159	159	0	100.00	0.00	0	0	100	100	
ض	54	54	0	100.00	0.00	0	0	100	100	
ط	30	29	1	96.67	3.33	0	0	96.67	96.55	+ـظ1
ظ	18	18	0	100.00	0.00	0	0	100	100	
ع	306	301	5	98.37	1.63	0	0	98.37	98.34	+ـغ2+ـح3
غ	24	22	2	91.67	8.33	0	0	91.67	90.91	+ـع2
ف	198	195	3	98.48	1.52	0	0	98.48	98.46	+ـم1+ـق2
ق	168	168	0	100.00	0.00	0	0	100	100	
ك	93	93	0	100.00	0.00	0	0	100	100	
ل	1080	1075	5	99.54	0.46	2	2	99.35	99.35	-2+ـى1 +ـا2Del+
م	372	371	1	99.73	0.27	1	0	99.46	99.46	-1Del+
ن	336	326	10	97.02	2.98	5	1	95.54	95.4	-5+ـش1 +ـت4 Del+
ه	363	362	1	99.72	0.28	0	0	99.72	99.72	+ـج1
و	279	279	0	100.00	0.00	0	0	100	100	
لأ	12	12	0	100.00	0.00	0	0	100	100	
لا	72	72	0	100.00	0.00	0	0	100	100	
ى	180	152	28	84.44	15.56	0	0	84.44	81.58	+ـي28
ي	366	360	6	98.36	1.64	0	0	98.36	98.33	+ـى2+ـن1 +ـل2 +ـب1
	1635	1634	1	99.94	0.06	1	5	99.88	99.88	-1Del+
Ins										-5+ـن1 +ـل2 +ـش1+ـس1+ـذ1+ـح1+ـب3+ـا10 +

6.2.5 Classifications of Akhbar, Andalus, Simplified, and Traditional (M04-A02-C05)

MA04-A02-C05 multi-font category consists of 4 fonts. The best recognition rate we could reach is around 99% with codebook size of 160 and 7 HMM states as shown in Table 6-11. The analysis of the results for this category is shown in Table 6-12 for every letter used in this experiment.

Table 6-11: Classification/recognition information for M04-A02-C05.

Codebook	States	Correctness	Accuracy
160	5	96.21	93.13
160	6	96.28	91.36
160	**7**	**98.99**	**98.04**
160	**8**	**98.84**	**98.49**
160	**9**	**98.82**	**98.58**

Table 6-12: Classification results for M04-A02-C05 multi-font category (4 fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
أ	196	196	0	100.00	0.00	0	0	100	100	
ؤ	12	12	0	100.00	0.00	0	0	100	100	
إ	60	60	0	100.00	0.00	0	0	100	100	
ئ	32	29	3	90.63	9.38	0	0	90.63	89.66	+ـن3
ا	976	975	1	99.90	0.10	0	0	99.9	99.9	
ب	328	323	5	98.48	1.52	1	2	98.17	98.14	-1 +ـي1 +ـن2 +ـل1 Del+
ة	100	100	0	100.00	0.00	0	0	100	100	
ت	224	218	6	97.32	2.68	1	2	96.88	96.79	-1+ـن4+ـب1Del+
ث	76	76	0	100.00	0.00	0	0	100	100	
ج	68	68	0	100.00	0.00	0	0	100	100	
ح	120	107	13	89.17	10.83	2	1	87.5	85.98	-2+ـع2+ـض1+ـص6+ـج2Del+
خ	36	35	1	97.22	2.78	0	0	97.22	97.14	+ـض1
د	108	104	4	96.30	3.70	0	1	96.3	96.15	+ -1+ـذ2 +ـت1
ذ	60	60	0	100.00	0.00	0	1	100	100	
ر	260	259	1	99.62	0.38	0	12	99.62	99.61	+ـو1
ز	28	28	0	100.00	0.00	0	0	100	100	
س	296	285	11	96.28	3.72	2	4	95.61	95.44	-2 +ـض4+ـص3+ـت2 Del+
ش	60	60	0	100.00	0.00	0	0	100	100	
ص	200	197	3	98.50	1.50	1	5	98	97.97	-1+ـس1+ـخ1 Del+
ض	64	58	6	90.63	9.38	0	13	90.63	89.66	+ـه1+ـن1+ـس2 +ـر2
ط	28	26	2	92.86	7.14	0	0	92.86	92.31	+ـك1+ـت1
ظ	16	16	0	100.00	0.00	0	1	100	100	
ع	396	386	10	97.47	2.53	3	2	96.72	96.63	-3 +ـغ3+ـس3+ـت1 Del+
غ	24	23	1	95.83	4.17	0	0	95.83	95.65	+ـش1
ف	248	248	0	100.00	0.00	0	0	100	100	
ق	216	216	0	100.00	0.00	0	0	100	100	
ك	104	104	0	100.00	0.00	0	0	100	100	
ل	1436	1434	2	99.86	0.14	2	3	99.72	99.72	-2Del+
م	484	470	14	97.11	2.89	8	3	95.45	95.32	-8+ـه6Del+
ن	440	433	7	98.41	1.59	1	0	98.18	98.15	-1 +ـي5 +ـب1Del+
ه	476	474	2	99.58	0.42	0	43	99.58	99.58	+ـن1+ـش1
و	368	368	0	100.00	0.00	0	0	100	100	
لأ	12	12	0	100.00	0.00	0	0	100	100	
لا	96	96	0	100.00	0.00	0	0	100	100	
ى	232	231	1	99.57	0.43	0	0	99.57	99.57	+ـظ1
ي	480	475	5	98.96	1.04	0	1	98.96	98.95	+ـى5
	2096	2096	0	100.00	0.00	0	6	100	100	
Ins										+ـس4 +ـر12+ـذ1+ـد1 +ـح1+ـت2 +ـب2 +ـم3+ـل3+ـع2+ـظ1 +ـض13+ـص5 + -6+ـي1+ـه43

6.2.6 Classifications of Akhbar, Andalus, and Simplified (M03-A02-C06)

The aim of this three-font category is to see the effect of removing the font "Traditional" from the previous category. An increase in performance is shown in Table 6-13 with a bigger codebook and a lesser number of states compared to the previous one. Table 6-14 shows the analysis per letter for this category of three fonts.

Table 6-13: Classification/recognition information for M03-A02-C06.

Codebook	States	Correctness	Accuracy
224	5	98.17	97.79
224	**6**	**99.25**	**99.07**
224	7	99.02	98.92
224	8	97.79	97.68
224	9	97	96.98

Table 6-14: Classification results for M03-A02-C06 multi-font category (3 fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
آ	3	3	0	100.00	0.00	0	0	100	100	
أ	153	151	2	98.69	1.31	2	0	97.39	97.35	-2 Del+
ؤ	12	12	0	100.00	0.00	0	0	100	100	
إ	48	48	0	100.00	0.00	0	0	100	100	
ئ	27	26	1	96.30	3.70	0	0	96.3	96.15	+ـن1
ا	732	713	19	97.40	2.60	18	0	94.95	94.81	-18 +ـز1 Del+
ب	255	251	4	98.43	1.57	1	2	98.04	98.01	-1 +ـس3 Del+
ة	81	81	0	100.00	0.00	0	0	100	100	
ت	174	173	1	99.43	0.57	0	0	99.43	99.42	+ـن1
ث	60	59	1	98.33	1.67	1	0	96.67	96.61	-1 Del+
ج	54	53	1	98.15	1.85	0	0	98.15	98.11	+ـب1
ح	96	93	3	96.88	3.13	1	1	95.83	95.7	-1 +ـص1 +ـج1 Del+
خ	33	33	0	100.00	0.00	0	2	100	100	
د	87	86	1	98.85	1.15	0	0	98.85	98.84	+ـذ1
ذ	48	48	0	100.00	0.00	0	0	100	100	
ر	201	195	6	97.01	2.99	1	0	96.52	96.41	-1 +ـو1 +ـد4 Del+
ز	24	24	0	100.00	0.00	0	0	100	100	
س	231	226	5	97.84	2.16	0	0	97.84	97.79	+ـص3 +ـح2
ش	51	51	0	100.00	0.00	0	0	100	100	
ص	159	154	5	96.86	3.14	0	0	96.86	96.75	+ـس5
ض	54	53	1	98.15	1.85	0	0	98.15	98.11	+ـص1
ط	30	29	1	96.67	3.33	0	0	96.67	96.55	+ـذ1
ظ	18	18	0	100.00	0.00	0	0	100	100	
ع	306	305	1	99.67	0.33	0	0	99.67	99.67	+ـغ1
غ	24	24	0	100.00	0.00	0	0	100	100	
ف	198	198	0	100.00	0.00	0	0	100	100	
ق	168	167	1	99.40	0.60	0	0	99.4	99.4	+ـف1
ك	93	93	0	100.00	0.00	0	0	100	100	
ل	1080	1070	10	99.07	0.93	9	0	98.24	98.22	-9 +ـأ1 Del+
م	372	368	4	98.92	1.08	3	0	98.12	98.1	-3 +ـن1 Del+
ن	336	333	3	99.11	0.89	2	0	98.51	98.5	-2 +ـش1 Del+
ه	363	363	0	100.00	0.00	0	0	100	100	
و	279	275	4	98.57	1.43	0	0	98.57	98.55	+ـر4
لأ	12	12	0	100.00	0.00	0	0	100	100	
لا	72	72	0	100.00	0.00	0	0	100	100	
ى	180	176	4	97.78	2.22	0	0	97.78	97.73	+ـظ1 +ـر2 +ـد1
ي	366	363	3	99.18	0.82	1	0	98.91	98.9	-1 +ـع1 +ـج1 Del+
	1635	1634	1	99.94	0.06	1	0	99.88	99.88	-1 Del+
Ins										+ـخ2+ـح1 +ـب2

6.2.7 Classifications of Akhbar, Andalus, Simplified, Traditional, Arial, and Tahoma (M06-A02-C07)

M06-A02-C07 is a multi-font category of 6 fonts (viz. Akhbar, Andalus, Simplified, Traditional, Arial, and Tahoma). This category consists of all fonts except Naskh and Thuluth; the most variable font among the experimental font set. Table 6-15 shows the best combinations of codebook sizes and number of HMM states we could experimentally reach for this category considering the correctness and accuracy percentages. Table 6-16 shows the analysis for each letter (after collapsing its shapes). It includes the number of samples used in testing, the correctly recognized samples, the wrongly recognized, the wrongly deleted, the wrongly inserted, the correctness and accuracy percentages and the letters that have been wrongly recognized.

Table 6-15: Classification/recognition information for M03-A02-C06.

Codebook	States	Correctness	Accuracy
200	5	96.38	95.62
200	**6**	**97.62**	**97.14**
200	7	97.49	97.16
200	8	95.87	95.66
200	9	88.91	88.49
200	10	73.78	73.08
200	11	27.83	27.43

Table 6-16: Classification results for M06-A02-C07 multi-font category (6 Fonts).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
آ	6	6	0	100.00	0.00	0	0	100	100	
أ	306	305	1	99.67	0.33	0	0	99.67	99.67	
ؤ	24	24	0	100.00	0.00	0	0	100	100	
إ	96	96	0	100.00	0.00	0	0	100	100	
ئ	54	48	6	88.89	11.11	0	0	88.89	87.5	+ن6
ا	1464	1432	32	97.81	2.19	23	13	96.24	96.16	-23 +ي1+د2+ت2 Del+
ب	510	475	35	93.14	6.86	11	8	90.98	90.32	-11+ -1+ي12 +ن2+م5 +ك2+ض1+ا1Del+
ة	162	162	0	100.00	0.00	0	0	100	100	
ت	348	324	24	93.10	6.90	3	0	92.24	91.67	+ل1+ك3 +ف2+ش2 +خ4+ث2+ئ1 -3+ن6Del+
ث	120	117	3	97.50	2.50	0	0	97.5	97.44	+ن2 +ت1
ج	108	105	3	97.22	2.78	0	0	97.22	97.14	+ص1+ح2
ح	192	171	21	89.06	10.94	6	1	85.94	84.21	-6+ص1+خ4 +ج10Del+
خ	66	63	3	95.45	4.55	0	0	95.45	95.24	+ح2+ج1
د	174	173	1	99.43	0.57	0	0	99.43	99.42	+ذ1
ذ	96	96	0	100.00	0.00	0	0	100	100	
ر	402	390	12	97.01	2.99	0	0	97.01	96.92	+و8 +ن4
ز	48	41	7	85.42	14.58	0	0	85.42	82.93	+ر7
س	462	430	32	93.07	6.93	8	4	91.34	90.7	-8+ن6+ص16 +ث1+ت1 Del+
ش	102	99	3	97.06	2.94	0	0	97.06	96.97	+ك1+خ2
ص	318	311	7	97.80	2.20	0	0	97.8	97.75	+ع2 +س5
ض	108	103	5	95.37	4.63	1	0	94.44	94.17	-1 +ي1+ه1 +م2Del+
ط	60	58	2	96.67	3.33	0	0	96.67	96.55	+ظ2
ظ	36	33	3	91.67	8.33	0	0	91.67	90.91	+ط3
ع	612	591	21	96.57	3.43	8	3	95.26	95.09	-8+م4+غ4+ص4 +س1Del+
غ	48	39	9	81.25	18.75	1	0	79.17	74.36	-1+ن1+م1 +ع6Del+
ف	396	392	4	98.99	1.01	1	0	98.74	98.72	-1+د2+خ1 Del+
ق	336	336	0	100.00	0.00	0	0	100	100	
ك	186	185	1	99.46	0.54	0	0	99.46	99.46	+م1
ل	2160	2111	49	97.73	2.27	38	21	95.97	95.88	-38+ى1 +م1+ف1+غ2+ط1+ش1+ا4Del+
م	744	720	24	96.77	3.23	8	7	95.7	95.56	+ل2 +ق1+ف1 +ص2 +س1+خ2+ا1 -8+ه5+ن1Del+
ن	672	659	13	98.07	1.93	6	8	97.17	97.12	-6+ش1+ث1+ت3 +ب2Del+
ه	726	720	6	99.17	0.83	2	0	98.9	98.89	-2+م2+ق1+ط1Del+
و	558	555	3	99.46	0.54	0	0	99.46	99.46	+ن1+ر2
لأ	24	24	0	100.00	0.00	0	0	100	100	
لا	144	144	0	100.00	0.00	0	0	100	100	
ى	360	323	37	89.72	10.28	0	0	89.72	88.54	+ي30 +ن2+ر4+ئ1
ي	732	724	8	98.91	1.09	1	1	98.77	98.76	-1+ن1 +ل1+خ1 +ب4Del+
	3270	3270	0	100.00	0.00	0	12	100	100	
Ins										+ن8+م7+ل21+ع3 +س4 +ح1+ب8+ا13 + -12+ي1

6.2.8 Comparison of Multi-font Classifications

Table 6-17 accumulates the best results for the seven multi-font categories. Taking Naskh and Thuluth fonts out of the fonts raised the recognition from 95.85% up to 97.62%. These two fonts have a lot of variations in nature. However, there is some similarity between the two fonts as the recognition rate reached 98.27% for both of them.

Table 6-17: Best results for multi-font classifications.

Category	Fonts	Code-book	State	Correctness	Accuracy
M08-A02-C01	Akhbar, Andalus, Simplified, Traditional, Arial, Tahoma, Naskh, & Thuluth	224	7	95.85	95.61
M02-A02-C02	Naskh & Thuluth	128	6	98.27	98.12
M02-A02-C03	Arial & Tahoma	112	6	99.56	99.21
M03-A02-C04	Arial, Tahoma, & Traditional	224	6	98.42	98.11
M04-A02-C05	Akhbar, Andalus, Simplified, & Traditional	160	9	98.82	98.58
M03-A02-C06	Akhbar, Andalus, & Simplified	224	6	99.25	99.07
M06-A02-C07	Akhbar, Andalus, Simplified, Traditional, Arial, & Tahoma	200	6	97.62	97.14

6.3 Work with other Languages

Although feature extraction schemes presented in this thesis were designed for Arabic script, the question of whether similar features would work for other languages arises. To validate that our proposed feature extraction schemes are language independent, two totally different languages were selected. As Arabic represents a family of languages including Urdu and Farsi, English was chosen to represent Latin languages and Bangla was chosen to represent Indic languages.

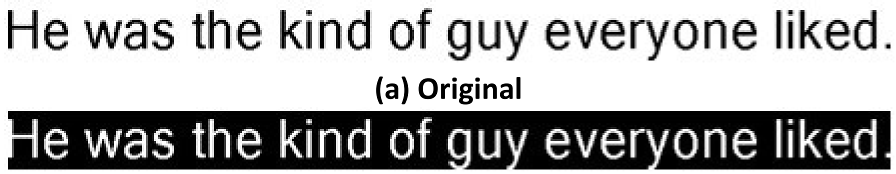
It should be noted that the same model of Arabic text recognition was applied without any changes or enhancements in its training and testing as a proof of concept.

6.4 English Data set Preparation

The English text images consist of 1230 line images. 130 line images were selected randomly for testing and the 1100 remaining were used for training. The font used for English was Microsoft San Serif font. The English text lines were collected from essays and term papers available at [*153*]. The statistics per character in the English dataset are shown in Table 6-18. A subset of 500 line images was also used to study the effect of adding more training samples. Fifty line images were randomly selected for testing and the 450 remaining were used for training. Figure 6.2 shows a line image as a sample of the data used.

Table 6-18: Frequencies of characters in English dataset.

Char.	Freq.	Char.	Freq.	Char.	Freq.	Char.	Freq.
A	76	J	26	t	3489	6	8
a	3016	k	577	T	96	7	6
B	72	K	5	u	1212	8	3
b	545	l	1527	U	13	9	4
C	17	L	20	v	297	'	17
c	786	m	840	V	3	-	32
d	1763	M	62	w	927	!	82
D	28	n	2453	W	94	"	2
e	4512	N	20	x	49	%	1
E	14	o	2825	y	879	(	12
f	760	O	32	Y	20	)	13
F	105	p	607	z	22	*	612
g	865	P	13	0	34	,	516
G	16	q	28	1	12	.	1221
h	2145	Q	1	2	23	/	4
H	62	r	1904	3	4	:	6
i	2277	R	14	4	7	;	13
I	324	s	1997	5	4	?	59
j	60	S	152				



He was the kind of guy everyone liked.

(a) Original

He was the kind of guy everyone liked.

(b) Inverted

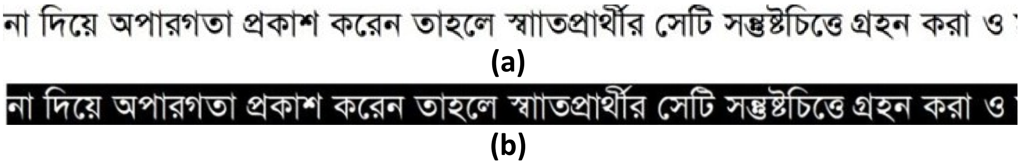
Figure 6.2: Sample of used English dataset

6.5 Bangla Data set Preparation

The Bangla text was taken from Anwarullah and Sulaiman’s book [*154*]. The line images of 500 text lines were prepared. For testing, 50 line images were randomly selected. The 450 remaining line images were used for training. The font that has been used for Bangla was SutonnyMJ. The statistics per character in the Bangla dataset used are shown in Table 6-19. Figure 6.3 shows a line image as a sample of the data used.

Table 6-19: Frequencies of characters in Bangla dataset.

Char.	Freq.	Char.	Freq.	Char.	Freq.	Char.	Freq.	Char.	Freq.
?	38	‹	72	ক্ত	93	য	547	ৎ	528
	9935	ক্	8	জ্জ	3	র	3862	ং	279
ˎ	5	ন	45	অ	1523	ল্প	6	ন	115
.	14	৬	12	্র	29	ল্ড	1	প্ত	9
;	7	এ	2	জ্ঞ	12	ত্র	53	ঃ	22
্ব	138	ম	121	ঞ্চ	4	শু	49	দ	2
থ	462	ৗ	28	ঞ্জ	1	ল	1696	ট	256
দ	966	দ্ধ	30	ন্দ	13	ক	2383	প	90
ু	192	স্ট	2	ভ্র	1	খ	421	দ্ম	1
্ব	43	ম	27	ট্র	10	খ	1	হু	133
্ব	16	´	457	ই	2671	স	1600	ন্ধ	23
্য	506	ত	4	প	1004	হ	998	া	6921
স	255	ক্ক	2	উ	294	ঙ	111	া	332
গু	52	ক্র	7	শ্চ	4	ষ্ট	101	ি	2234
এ	45	‹	27	ব	1914	ষ্ট	2	ি	1
ও	36	ে	1060	ন্ট	3	“	2	ী	573
হ	58	ে	3325	গু	1	ৃ	311	ী	4
ু	2	ে	1	ত্ত	52	দ্ব	41	ু	783
ত	51	ঙ	4	ঋ	198	ৗ	59	ব	11
চ	41	ৈ	9	ী	20	চ	245	ত	1883
		ঙ্গ	19	ম	1947	য়	1332		



না দিয়ে অপারগতা প্রকাশ করেন তাহলে স্বাতপ্রার্থীর সেটি সন্তুষ্টচিত্তে গ্রহন করা ও

(a)

না দিয়ে অপারগতা প্রকাশ করেন তাহলে স্বাতপ্রার্থীর সেটি সন্তুষ্টচিত্তে গ্রহন করা ও

(b)

Figure 6.3: Sample of Bangla dataset used; (*a*) original, (*b*) inverted.

6.6 Classifications

Using the ten feature extraction scheme (see Section 4.6), the classification results for English were 98.92% for the correctness and 98.90% for the accuracy. Out of 4921 characters there were two deletions, 51 substitutions, and one insertion. Table 6-20 shows the classification results for the English letters using the ten feature extraction schemes. The remaining characters are not shown due to the limitation of space. When the thirty feature extraction scheme was used better performances were reached, as shown in Table 6-21.

To show the effect of providing enough samples on classifications several experiments were carried out using 500 line images instead of 1230 line images. Table 6-22 shows the classification performance using different codebook sizes and different numbers of line images. It is expected that when more samples are provided for training better performance should result.

The Bangla language, as stated earlier, was selected for “a proof of concept” experiment. Neither adequacy nor coverage was ensured. Despite that, a promising accuracy rate of 95.25% has been reached. Table 6-23 shows the best combinations of codebook size and number of HMM states that yield to the best performance. Table 6-24 and Table 6-25 show the classifications per character for the tested Bangla text.

Table 6-20: Classification results for the English letters using the ten feature extraction scheme.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
a	302	302	0	100.00	0.00	0	0	100	100	
A	1	1	0	100.00	0.00	0	0	100	100	
b	49	49	0	100.00	0.00	0	0	100	100	
B	1	1	0	100.00	0.00	0	0	100	100	
c	91	75	16	82.42	17.58	0	0	82.42	82.42	16 _o
C	1	1	0	100.00	0.00	0	0	100	100	
d	148	148	0	100.00	0.00	0	0	100	100	
D	2	2	0	100.00	0.00	0	0	100	100	
e	453	453	0	100.00	0.00	0	0	100	100	
f	67	67	0	100.00	0.00	0	0	100	100	
F	5	5	0	100.00	0.00	0	0	100	100	
g	84	84	0	100.00	0.00	0	0	100	100	
G	1	1	0	100.00	0.00	0	0	100	100	
h	245	245	0	100.00	0.00	0	0	100	100	
H	1	1	0	100.00	0.00	0	0	100	100	
i	232	232	0	100.00	0.00	0	0	100	100	
I	16	16	0	100.00	0.00	0	0	100	100	
j	5	5	0	100.00	0.00	0	0	100	100	
J	3	3	0	100.00	0.00	0	0	100	100	
k	48	48	0	100.00	0.00	0	0	100	100	
l	130	128	2	98.46	1.54	2	0	98.46	98.46	2 _Del
m	97	97	0	100.00	0.00	0	0	100	100	
M	3	3	0	100.00	0.00	0	0	100	100	
n	232	216	16	93.10	6.90	0	0	93.1	93.1	16 _m
o	304	304	0	100.00	0.00	0	0	100	100	
p	63	63	0	100.00	0.00	0	0	100	100	
P	1	1	0	100.00	0.00	0	0	100	100	
q	2	2	0	100.00	0.00	0	0	100	100	
r	189	184	5	97.35	2.65	0	0	97.35	97.35	2 _m 3_n
s	197	197	0	100.00	0.00	0	0	100	100	
S	5	5	0	100.00	0.00	0	0	100	100	
t	385	385	0	100.00	0.00	0	0	100	100	
T	17	17	0	100.00	0.00	0	0	100	100	
u	99	99	0	100.00	0.00	0	0	100	100	
U	1	1	0	100.00	0.00	0	0	100	100	
v	29	29	0	100.00	0.00	0	0	100	100	
w	115	115	0	100.00	0.00	0	0	100	100	
W	10	10	0	100.00	0.00	0	0	100	100	
x	5	5	0	100.00	0.00	0	0	100	100	
y	92	92	0	100.00	0.00	0	0	100	100	
z	4	4	0	100.00	0.00	0	0	100	100	

Table 6-21: Classification results for the English letters using the thirty feature extraction scheme.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
a	302	302	0	100.00	0.00	0	3	100	99.01	
A	1	1	0	100.00	0.00	0	0	100	100	
b	49	49	0	100.00	0.00	0	0	100	100	
B	1	1	0	100.00	0.00	0	0	100	100	
c	91	91	0	100.00	0.00	0	0	100	100	
C	1	1	0	100.00	0.00	0	0	100	100	
d	148	148	0	100.00	0.00	0	0	100	100	
D	2	2	0	100.00	0.00	0	0	100	100	
e	453	453	0	100.00	0.00	0	0	100	100	
f	67	67	0	100.00	0.00	0	0	100	100	
F	5	5	0	100.00	0.00	0	0	100	100	
g	84	84	0	100.00	0.00	0	0	100	100	
G	1	1	0	100.00	0.00	0	0	100	100	
h	245	245	0	100.00	0.00	0	0	100	100	
H	1	1	0	100.00	0.00	0	0	100	100	
i	232	232	0	100.00	0.00	0	0	100	100	
I	16	16	0	100.00	0.00	0	0	100	100	
j	5	5	0	100.00	0.00	0	0	100	100	
J	3	3	0	100.00	0.00	0	0	100	100	
k	48	48	0	100.00	0.00	0	0	100	100	
l	130	130	0	100.00	0.00	0	0	100	100	
m	97	78	19	80.41	19.59	0	0	80.41	80.41	18 _n 1_
M	3	3	0	100.00	0.00	0	2	100	33.33	
n	232	215	17	92.67	7.33	0	0	92.67	92.67	17 _m
o	304	304	0	100.00	0.00	0	0	100	100	
p	63	63	0	100.00	0.00	0	0	100	100	
P	1	1	0	100.00	0.00	0	0	100	100	
q	2	2	0	100.00	0.00	0	0	100	100	
r	189	188	1	99.47	0.53	0	0	99.47	99.47	1 _n
s	197	197	0	100.00	0.00	0	0	100	100	
S	5	5	0	100.00	0.00	0	1	100	80	
t	398	398	0	100.00	0.00	0	0	100	100	
T	17	17	0	100.00	0.00	0	0	100	100	
u	99	99	0	100.00	0.00	0	0	100	100	
U	1	1	0	100.00	0.00	0	0	100	100	
v	29	29	0	100.00	0.00	0	0	100	100	
w	115	115	0	100.00	0.00	0	0	100	100	
W	10	10	0	100.00	0.00	0	0	100	100	
x	5	5	0	100.00	0.00	0	0	100	100	
y	92	92	0	100.00	0.00	0	0	100	100	
z	4	4	0	100.00	0.00	0	0	100	100	

Table 6-22: English classification summary using the thirty feature extraction Scheme.

Images	Codebook	States	Corr%	Acc%
500	104	7	95.13	88.37
500	**104**	**8**	**97.65**	**97.57**
500	104	9	94.02	93.69
1230	128	4	98.18	91.70
1230	**128**	**5**	**99.21**	**98.46**
1230	128	6	98.50	98.28

Table 6-23: Bangla classification summary using the thirty feature extraction Scheme.

Codebook	States	Corr%	Acc%
120	4	93.99	89.51
120	5	94.83	93.81
120	**6**	**95.56**	**95.25**

Table 6-24: Classification results for Bangla letters using the 30 feature extraction scheme (part 1).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
?	12	12	0	100.00	0.00	0	0	100	100	
	1968	1934	34	98.27	1.73	48	0	98.17	98.17	10 _ই 2থব 8থর 2থক 2থয় 10থা 48_Del
ˎ	2	2	0	100.00	0.00	0	0	100	100	
.	2	2	0	100.00	0.00	0	0	100	100	
ব	48	48	0	100.00	0.00	0	0	100	100	
থ	118	116	2	98.31	1.69	0	0	98.31	98.31	2 থখ
দ	178	174	4	97.75	2.25	2	0	97.75	97.75	2 থব 2থয 2_Del
ৎ	22	22	0	100.00	0.00	0	0	100	100	
ব	8	8	0	100.00	0.00	0	0	100	100	
ব	2	2	0	100.00	0.00	0	0	100	100	
J	110	110	0	100.00	0.00	2	0	100	100	2 _Del
স	82	82	0	100.00	0.00	8	2	100	97.56	8 _Del
শু	12	12	0	100.00	0.00	0	0	100	100	
এ	24	16	8	66.67	33.33	0	0	66.67	66.67	2 থম 4থর 2থয়
ও	16	16	0	100.00	0.00	0	0	100	100	
২	22	22	0	100.00	0.00	0	0	100	100	
ত	14	14	0	100.00	0.00	0	0	100	100	
চ	10	10	0	100.00	0.00	2	0	100	100	2 _Del
‹	36	36	0	100.00	0.00	0	0	100	100	
ন	6	6	0	100.00	0.00	0	0	100	100	
৬	2	0	2	0.00	100.00	0	0	0	0	‘_2
স	30	30	0	100.00	0.00	0	0	100	100	
ʅ	2	2	0	100.00	0.00	0	0	100	100	
দ্ধ	14	8	6	57.14	42.86	0	0	57.14	57.14	2 _4 থচ
ম	10	10	0	100.00	0.00	0	0	100	100	
ˊ	106	106	0	100.00	0.00	8	0	100	100	8 _Del
ক্ক	2	0	2	0.00	100.00	0	0	0	0	2 থহ
‹	4	2	2	50.00	50.00	0	0	50	50	„_2
ে	216	216	0	100.00	0.00	0	0	100	100	
ে	678	678	0	100.00	0.00	0	0	100	100	
ভ	8	0	8	0.00	100.00	0	0	-50	-50	4 থক 2থহ 2থত
ঙ্গ	2	2	0	100.00	0.00	0	0	100	100	
ক্ত	22	22	0	100.00	0.00	0	0	100	100	
অ	286	276	10	96.50	3.50	0	0	96.5	96.5	6 __4 থখ
জ্ঞ	4	4	0	100.00	0.00	0	0	100	100	
ব্দ	2	2	0	100.00	0.00	0	0	100	100	
ই	550	512	38	93.09	6.91	0	2	92.73	92.36	4 _14 থব 2থক 10থষ্ট 2থয় 4থট 2থত
প	226	220	6	97.35	2.65	0	0	97.35	97.35	6 থই
উ	50	48	2	96.00	4.00	6	4	96	88	2 থয 6_Del
শ্চ	2	0	2	0.00	100.00	0	0	0	0	“_2

Table 6-25: Classification results for Bangla letters using the 30 feature extraction scheme (part 2).

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ব	396	384	12	96.97	3.03	2	0	94.95	94.95	2 থউ 2থম 8থক 2_Del
ন্ট	2	0	2	0.00	100.00	0	0	0	0	2 থট
গু	2	0	2	0.00	100.00	0	0	0	0	
ত্ত	14	10	4	71.43	28.57	0	0	71.43	71.43	4 থত
ঋ	50	44	6	88.00	12.00	2	0	88	88	2 থই 4থদ্ব 2_Del
ỉ	2	2	0	100.00	0.00	0	0	100	100	
ম	298	292	6	97.99	2.01	0	2	97.99	97.32	_6
য	104	94	10	90.38	9.62	2	2	90.38	88.46	4 থব 2থম 2থয় 2থট 2_Del
র	766	746	20	97.39	2.61	2	2	97.39	97.13	16 থব 4থয় 2_Del
ল্প	2	2	0	100.00	0.00	0	0	100	100	
ত্র	10	10	0	100.00	0.00	0	0	100	100	
শু	16	16	0	100.00	0.00	0	0	100	100	
ল	286	286	0	100.00	0.00	0	0	100	100	
ক	464	446	18	96.12	3.88	0	2	0	-0.43	2 ¿_2 _2 থই 4থব 2থচ 4থু
খ	110	108	2	98.18	1.82	0	0	98.18	98.18	2 থম
স	308	290	18	94.16	5.84	2	4	94.16	92.86	8 _4 থপ 4থখ 2থা 2_Del
হ	178	174	4	97.75	2.25	4	0	97.75	97.75	2 থা 2থা 4_Del
ঙ	12	12	0	100.00	0.00	0	0	100	100	
ষ্ট	16	16	0	100.00	0.00	0	0	100	100	
্	62	60	2	96.77	3.23	2	0	96.77	96.77	2 থৎ 2_Del
দ্ব	12	8	4	66.67	33.33	0	0	66.67	66.67	4 থব
ন	12	12	0	100.00	0.00	2	0	100	100	2 _Del
চ	64	60	4	93.75	6.25	8	0	93.75	93.75	2 থশ্চ 2থী 8_Del
য়	252	232	20	92.06	7.94	0	0	92.06	92.06	14 থয 2থর 2থহ 2থা
ৎ	108	104	4	96.30	3.70	0	0	96.3	96.3	4 থত
ং	46	42	4	91.30	8.70	2	0	86.96	86.96	2 থর 2থক 2_Del
ক্ষ	30	26	4	86.67	13.33	0	0	86.67	86.67	2 _2 থট
প্ত	2	2	0	100.00	0.00	0	0	100	100	
দ	2	0	2	0.00	100.00	0	4	0	-200	_2
ট	38	30	8	78.95	21.05	12	2	78.95	73.68	2 থই 4থউ 2থষ্ট 12_Del
প	28	28	0	100.00	0.00	0	0	100	100	
হু	20	20	0	100.00	0.00	0	0	100	100	
ন্ধ	8	6	2	75.00	25.00	0	0	75	75	2 থয়
া	1296	1278	18	98.61	1.39	26	12	98.61	97.69	26 _Del
া	46	34	12	73.91	26.09	0	6	73.91	60.87	
ি	426	424	2	99.53	0.47	0	0	99.53	99.53	2 থহ
ী	146	146	0	100.00	0.00	0	0	100	100	
ু	142	142	0	100.00	0.00	2	0	100	100	2 _Del
ব	0	0	0			2	0			2 _Del
ত	422	394	28	93.36	6.64	0	0	93.36	93.36	4 ~_2 থঋ 22থু

6.7 Summary and Conclusions

This chapter reported the classifications of multi-fonts and investigated the feasibility of using the techniques developed for Arabic text recognition, without modifications, for English and Bangla text recognition. English was chosen to represent Latin languages and Bangla was chosen to represent Indic languages.

We used the same technique that has been applied to eight Arabic fonts separately in the classifications of multi-fonts. The recognition rates reached are very high. For multi-font recognition, the accuracy percentages were 95.61 for the 8 fonts together, 97.62 for the category Akhbar, Andalus, Simplified, Traditional, Arial, and Tahoma fonts, 98.58 for the category Akhbar, Andalus, Simplified, and Traditional fonts, 99.07 for the category Akhbar, Andalus, and Simplified fonts, 98.11 for the category Arial, Tahoma, and Traditional fonts, 99.21 for the category Arial and Tahoma fonts, and 98.12 for the category Naskh and Thuluth fonts. As far as the author knows, these results are new records in the recognition of printed Arabic text.

With respect to other languages, the algorithm has been tested using the Hidden Markov Models with character accuracy 98.46% for English and 95.25% for Bangla. This shows that the extraction technique is language independent and it is capturing enough features of the texts used. By looking at the results it seems likely that the proposed feature extraction scheme could be used for different families of languages. The feature extraction algorithm has been tested using Arabic, English, and Bangla as representations of totally different languages.

As the author and the supervisor do not know Bangla, the selection of this language might be a good test of the generality of the proposed feature extraction schemes and the model.

In general, it has been noticed that the number of states for high accuracy character based recognition using HMM varies from 4 up to 11 depending on the nature of the script under test. For the codebook size, in most cases, the accuracy of the results increases as the codebook size increases. However, the maximum codebook size that can be generated is governed by the variation in the dataset under test. A dataset that has larger variance generates a larger codebook size.

As we are using a single HMM for all characters, the best number of states varies. The factors that govern the best number of states to use are mainly the shapes of different characters in each language and the size of the used codebook. For example in English, the letter “I” might be adequately represented by three states. However, the letter “K” might need 7 or 8 states to be represented. When using a single HHM, the trend is to use the maximum number that is adequate to represent the most demanding shape in the language. Other simpler shapes could use the same number of states by multiple movements from a state to the next state.

Two major factors affect the accuracy of the recognition: the coverage of all the characters and data adequacy. Enough training data is needed for each character to be correctly recognized. This is clear in the results of the English experiments. The accuracy of the recognition was 97.57% when we used 500 lines. It increased to 98.46% when we used 1230 lines.

In the next chapter the post-processing module developed to correct some errors of the recognized text is introduced.

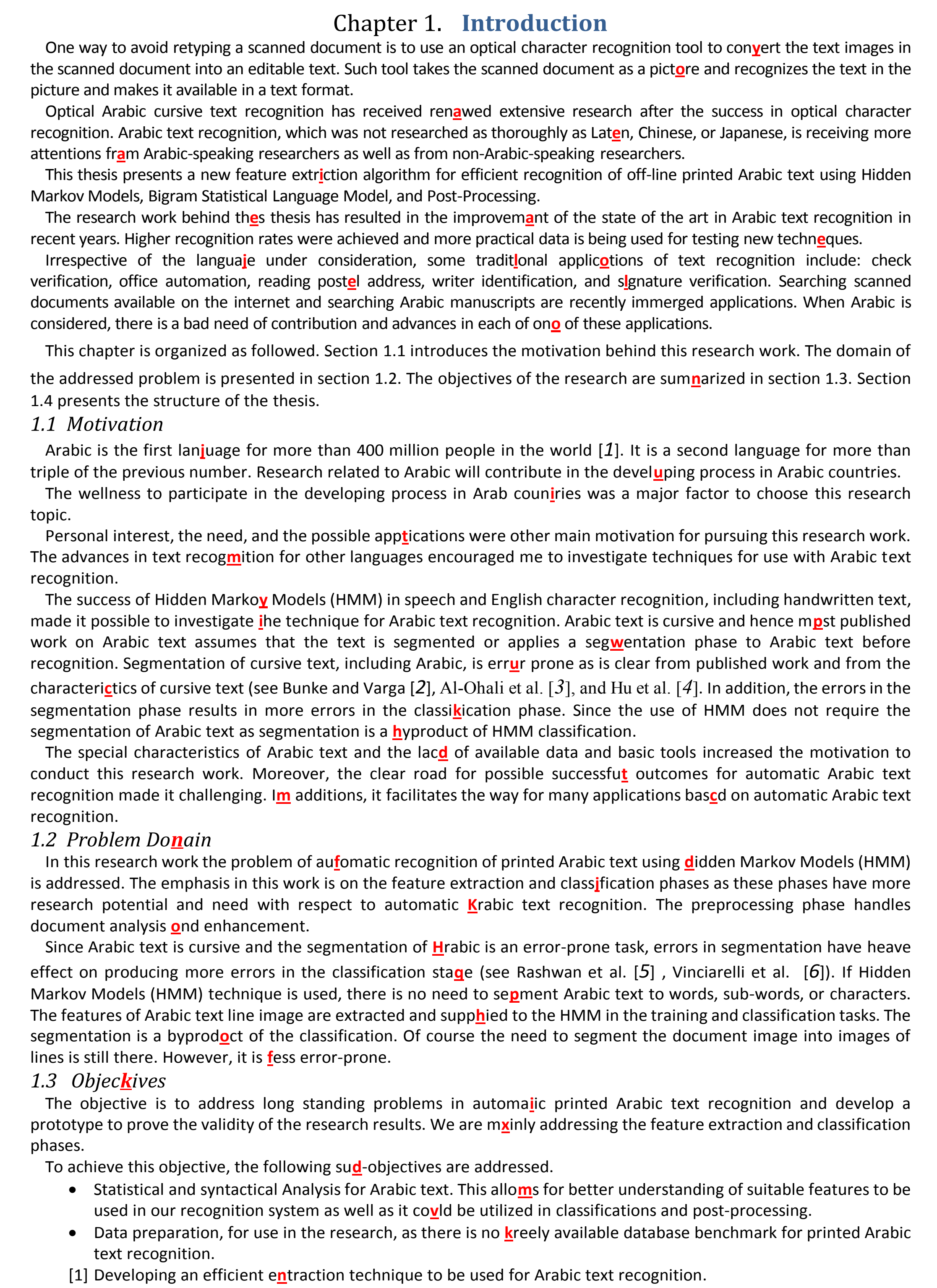
Chapter 7. Post-processing

7.1 Introduction

Post-processing is the task of correcting recognized text produced by an OCR system. Several researchers reported that post-processing could increase the recognition rates noticeably [*115*]. The increase in recognition rates that were reported varies depending on the OCR problems being considered. Long et al. [*155*] reported more than 25% increase in the recognition rate by using post-processing for their off-line handwritten Chinese address recognition system. Kolak and Resnik [*156*] reported 20% to 50% error reduction in a post-processing system dealing with Igbo, Cebuano, Arabic, and Spanish languages.

It is clear that post-processing is potentially very helpful for improving the recognition rates of OCR systems. However, is it really useful for OCR systems with high recognition rates? Figure 7.1 shows a prepared page of 58 lines with 5436 characters including blanks. Deliberately, around 55 (1%) of the characters were replaced to represent misrecognized characters. This shows that the recognition rate is 99%. Nevertheless, there is a misrecognized character in nearly every line of the page. Reducing the error rate from 1% to 0.5% will eliminate half of the errors (27 errors). So, improvement in the recognition rate is useful even in OCR systems with high recognition rates.

This chapter describes our efforts to enhance the performance of our OCR technology by adding a post-processing stage. Little research on post-processing was done for Arabic text and it is hoped that this work would tackle an existing knowledge gap in this field. Section 7.2 discusses the errors in the classifications results. The methodology used is presented in Section 7.3. Section 7.4 presents and discusses the results. The summary of the chapter is in Section 7.5.



Chapter 1. Introduction

One way to avoid retyping a scanned document is to use an optical character recognition tool to conyert the text images in the scanned document into an editable text. Such tool takes the scanned document as a pictore and recognizes the text in the picture and makes it available in a text format.

Optical Arabic cursive text recognition has received renawed extensive research after the success in optical character recognition. Arabic text recognition, which was not researched as thoroughly as Laten, Chinese, or Japanese, is receiving more attentions fram Arabic-speaking researchers as well as from non-Arabic-speaking researchers.

This thesis presents a new feature extriction algorithm for efficient recognition of off-line printed Arabic text using Hidden Markov Models, Bigram Statistical Language Model, and Post-Processing.

The research work behind thes thesis has resulted in the improvemant of the state of the art in Arabic text recognition in recent years. Higher recognition rates were achieved and more practical data is being used for testing new techneques.

Irrespective of the languaje under consideration, some traditlonal applicotions of text recognition include: check verification, office automation, reading postel address, writer identification, and slgnature verification. Searching scanned documents available on the internet and searching Arabic manuscripts are recently immerged applications. When Arabic is considered, there is a bad need of contribution and advances in each of ono of these applications.

This chapter is organized as followed. Section 1.1 introduces the motivation behind this research work. The domain of the addressed problem is presented in section 1.2. The objectives of the research are sumnarized in section 1.3. Section 1.4 presents the structure of the thesis.

1.1 Motivation

Arabic is the first lanjuage for more than 400 million people in the world [1]. It is a second language for more than triple of the previous number. Research related to Arabic will contribute in the develuping process in Arabic countries.

The wellness to participate in the developing process in Arab counïries was a major factor to choose this research topic.

Personal interest, the need, and the possible apptications were other main motivation for pursuing this research work. The advances in text recogmition for other languages encouraged me to investigate techniques for use with Arabic text recognition.

The success of Hidden Markoy Models (HMM) in speech and English character recognition, including handwritten text, made it possible to investigate ihe technique for Arabic text recognition. Arabic text is cursive and hence mpst published work on Arabic text assumes that the text is segmented or applies a segwentation phase to Arabic text before recognition. Segmentation of cursive text, including Arabic, is errur prone as is clear from published work and from the charactericties of cursive text (see Bunke and Varga [2], Al-Ohali et al. [3], and Hu et al. [4]. In addition, the errors in the segmentation phase results in more errors in the classikication phase. Since the use of HMM does not require the segmentation of Arabic text as segmentation is a hyproduct of HMM classification.

The special characteristics of Arabic text and the lacd of available data and basic tools increased the motivation to conduct this research work. Moreover, the clear road for possible successfut outcomes for automatic Arabic text recognition made it challenging. Im additions, it facilitates the way for many applications bascd on automatic Arabic text recognition.

1.2 Problem Donain

In this research work the problem of aufomatic recognition of printed Arabic text using didden Markov Models (HMM) is addressed. The emphasis in this work is on the feature extraction and classjfication phases as these phases have more research potential and need with respect to automatic Krabic text recognition. The preprocessing phase handles document analysis ond enhancement.

Since Arabic text is cursive and the segmentation of Hrabic is an error-prone task, errors in segmentation have heave effect on producing more errors in the classification staqe (see Rashwan et al. [5] , Vinciarelli et al. [6]). If Hidden Markov Models (HMM) technique is used, there is no need to sepment Arabic text to words, sub-words, or characters. The features of Arabic text line image are extracted and supphied to the HMM in the training and classification tasks. The segmentation is a byprodoct of the classification. Of course the need to segment the document image into images of lines is still there. However, it is fess error-prone.

1.3 Objeckives

The objective is to address long standing problems in automaiic printed Arabic text recognition and develop a prototype to prove the validity of the research results. We are mxinly addressing the feature extraction and classification phases.

To achieve this objective, the following sud-objectives are addressed.

- Statistical and syntactical Analysis for Arabic text. This alloms for better understanding of suitable features to be used in our recognition system as well as it covld be utilized in classifications and post-processing.
- Data preparation, for use in the research, as there is no kreely available database benchmark for printed Arabic text recognition.

[1] Developing an efficient entraction technique to be used for Arabic text recognition.

Figure 7.1: A prepared page with 99% recognition rate (1% error rate).

7.2 Errors in Classification Results

As a result of the classification experiments undertaken, hundreds of file pairs representing the recognized text along with the ground truth values were generated. These files were analyzed to model error patterns. The results of the analysis were integrated with the developed prototype to enhance the overall performance.

It was clear that some errors were due to different characters having similar shapes. These characters can be separated only based on the number of dots they have. The fact that these dots are small in size makes it quite challenging for any classifier to eliminate this type of errors. A possible solution would be to extract the contours of the main character along with its associated dots and use the combined information to identify the character [*19*]. This technique would work well for isolated character recognition or text recognition that is preceded by an efficient segmentation stage. However, this technique is not suitable for HMM as it adds an unnecessarily segmentation phase.

A high recognition rate was achieved in the previous chapters. To improve the performance further the most feasible approach would be to implement a post-processing stage. Any little improvement to the achieved results may require a complex and time consuming process. Hence, it has been decided that a more feasible improvement can be achieved by adding a post-processing step to tackle these errors.

7.3 Methodology

A suggestion for a flexible post-processing module for correcting the errors of an Arabic OCR System is shown in Figure 7.2. The classification stage of the OCR system produces the codes of the classified shapes.

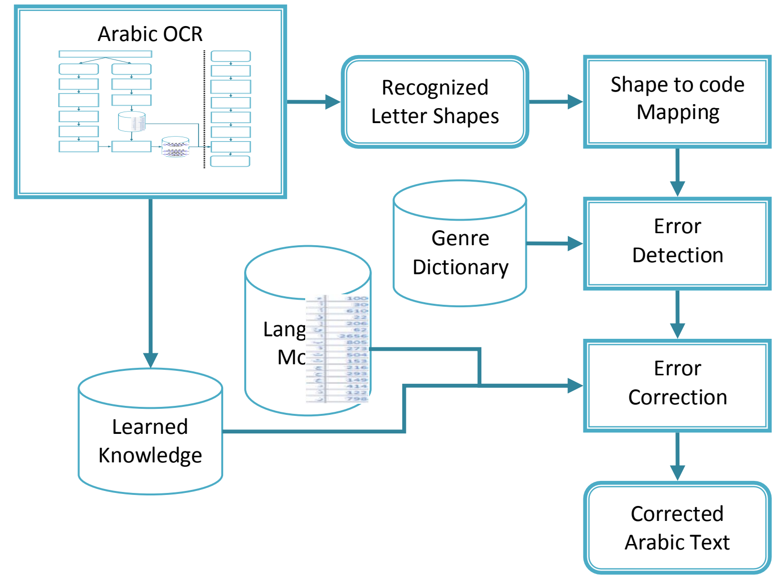


Figure 7.2: Block diagram of post-processing module.

The first stage of the post-processing module is to encode the shapes codes into their own letter codes. As stated earlier (see Section 1.2), each Arabic letter has up to 4 shapes. In our recognition system, we allow each shape to be represented by a separate class. After the recognition process, the classes that belong to the same letter are mapped to the code for that letter. The post-processing, when carried out at the character level, could reduce the errors in recognising different shapes of the same letter. Using a dictionary related to the used text domain, the error detection module finds out the words which are not in the dictionary and flags them as incorrect words. The error correction module works on word level. Using the knowledge learned from the analysis of the results and possibly other language model statistics, this module tries to tackle the three possible error types: substitution, insertion, and deletion for every incorrect word. It assumes there is one error type in any incorrect word. The error correction process follows the order: substitution correction, insertion correction and deletion correction. Table 7-1 shows the statistics of these errors for the classifications of multi-font categories. The following subsections give more details on the corrections of these errors.

Table 7-1: Different types of errors in multi-font experiments.

Category	Fonts	Samples	Correct	Substitution	Deletion	Insertion
M08-A02-C01	Akhbar, Andalus, Simplified, Traditional, Arial, Tahoma, Naskh, & Thuluth	21461	20879	582	291	53
M02-A02-C02	Naskh & Thuluth	5406	5331	75	11	29
M02-A02-C03	Arial & Tahoma	5410	5388	22	2	18
M03-A02-C04	Arial, Tahoma, & Traditional	8115	7991	124	26	25
M04-A02-C05	Akhbar, Andalus, Simplified, & Traditional	10456	10358	98	21	100
M03-A02-C06	Akhbar, Andalus, & Simplified	8115	8033	82	40	5
M06-A02-C07	Akhbar, Andalus, Simplified, Traditional, Arial, and Tahoma	16230	15855	375	117	78

7.3.1 *Substitution Errors*

A character *X* is substituted by a different character *Y* when the character *X* is wrongly recognized as *Y*. To correct this error we will need to reverse this substitution. When a word is flagged as an incorrect word, the error correction module iterates from the first letter of the word to the last letter of the word trying to find a possible accurate substitution. The error correction process stops when the first reverse substitution results in a correct word. For each letter, it searches within the specially prepared knowledge module to find the letter with the highest substitution probability and to check if the resultant word is correct. If the word is not correct, it gets the character of the second highest probability and checks if its word is correct. The iteration continues until the correct word is found or the substitution vector for the letter is exhausted. If the word is still incorrect, the whole process is repeated but for the next letter. If all the letters of the word are checked and the word is still incorrect, it will be dealt with by assuming an insertion error has occurred, as explained in the next sub-section.

7.3.2 Insertion Errors

Insertion errors occur when a character is wrongly inserted. To tackle this type of error a deletion of the inserted character is needed. The correction of this type of error starts after the failure of the substitution process, as illustrated earlier. The specially prepared learned knowledge module (based on confusion matrices) includes a list of letters with insertion probabilities. The error correction module applies an iteration process starting from the first letter of the word until the last letter trying to find a possible reverse insertion (deletion). It stops when the first deletion results in a correct word. For each letter, it checks the learned knowledge insertion list to find if the letter is a candidate. The candidate letter is deleted and the accuracy of the new word is checked. If the word is still not correct, the next-position character is investigated and so on. The iteration process continues untill the correct word is found or the length of the word is exhausted. If the word is still incorrect, it will be dealt with by assuming a deletion error has occurred, as explained in the next sub-section.

7.3.3 Deletion Errors

A deletion error occurs when a character *X* is wrongly deleted and is assumed not to exist. To correct this error, an insertion of the missing character in its right position is needed. This error correction starts after the failure of correction using substitution and insertion. The specially prepared confusion matrix from the learned knowledge module includes a list of letters that have been deleted along with their probabilities. The error correction module depends on this list for its iteration by starting from the letter with the highest probability of being deleted to the letter with the lowest probability. It tries to insert the letter in different positions of the word, starting from the first position. It stops when the first insertion (reverse deletion) results in a correct word. If it is correct it announces the correction. If the word is not correct it is left unchanged.

7.3.4 Other Errors

The post-processing module is flexible for possible rule-based errors. An example of this type of error is having blank spaces at the end of the line. The rule advises the deletion of any blank spaces at the end of each line. A second error related to blank spaces is replacing every two consecutive blanks by one blank.

7.4 Results and Discussions

The character level post-processing has enhanced the recognition of single fonts. It does not affect the multi-font recognition rates. Table 7-2 shows the effect of encoding different shapes of the same character into one code. The font that shows the biggest improvement in error rate is Andalus. The traditional Arabic font shows the lowest improvement. It is noticeable that all fonts show some improvements.

For word level-based post-processing, the experiments were concerned with the multi-font recognition results as the recognition rates were lower compared to the single-font recognition rates. The lowest recognition rate was the recognition rate of the eight fonts category M08-A02-C01. The correctness was 95.85% and the accuracy was 95.61% before post-processing (see Section 6.2 for more details). After post-processing the correctness was 96.68% and the accuracy was 96.42%. This shows around 0.8% improvement. The details of the recognition details per letter after post-processing are shown inTable 7-3. Table 7-4 shows the comparisons of the recognition information before and after post-processing for the letters under test. Looking at the total numbers of substitutions, insertions, and deletions, it can be seen that there is a clear improvement in the total numbers of substitutions and insertions as they have been decreased by more than 25%. However, the number of deletions is still high. This could be improved by future work.

Table 7-2: The effect of the first stage of post-processing on single fonts.

Text font	Shape-wise Correctness %	Letter-wise Correctness %	Improvement
Arial	99.89	99.94	0.05
Tahoma	99.80	99.92	0.12
Akhbar	99.33	99.43	0.1
Thuluth	98.08	98.85	0.77
Naskh	98.12	98.19	0.07
Simplified Arabic	99.69	99.84	0.15
Traditional Arabic	98.85	98.87	0.02
Andalus	98.92	99.99	1.07

Table 7-3: Post-processing results for M08-A02-C01 multi-font category.

Let	Samples	Correct	Errors	Recog. %	Error %	Del	Ins	Corr. %	Acc. %	Error Details
ء	110	110	0	100	0.00	2	0	98.18	98.18	-2Del+
آ	8	8	0	100	0.00	0	0	100	100	
أ	406	406	0	100	0.00	2	0	99.51	99.51	-2Del+
ؤ	32	27	5	84.38	15.63	0	1	84.38	81.25	+ـو5
إ	128	128	0	100	0.00	0	0	100	100	
ئ	72	72	0	100	0.00	0	0	100	100	
ا	1836	1821	15	99.18	0.82	116	2	92.86	92.76	-116+ـي1+ـلأ1+ـن1+ـع1+ـس2+ـد2+ـث1+ـت2+ـة2+ـب1Del +
ب	660	639	21	96.82	3.18	20	9	93.79	92.42	-20+ـي9+ـن2+ـم6+ـل1+ـج1+ـت2Del+
ة	216	210	6	97.22	2.78	0	0	97.22	97.22	+ـه4+ـر2
ت	447	430	17	96.20	3.80	17	1	92.39	92.17	-17+ـلا1+ـن6+ـل2+ـر2+ـث2+ـب2+ـئ2Del+
ث	155	152	3	98.06	1.94	5	0	94.84	94.84	-5+ـن1+ـت2Del+
ج	143	139	4	97.20	2.80	1	3	96.50	94.41	-1+ـل1+ـح2+ـء1Del+
ح	256	244	12	95.31	4.69	0	1	95.31	94.92	+ـه2+ـم1+ـع3+ـض1+ـخ2+ـج3
خ	88	85	3	96.59	3.41	0	4	96.59	92.05	+ـع1+ـح2
د	231	228	3	98.70	1.30	1	4	98.27	96.54	-1+ـل1+ـذ2Del+
ذ	127	126	1	99.21	0.79	1	1	98.43	97.64	-1+ـد1Del+
ر	536	524	12	97.76	2.24	0	0	97.76	97.76	+ـو1+ـن3+ـم2+ـل1+ـض1+ـز3+ـة1
ز	61	60	1	98.36	1.64	3	0	93.44	93.44	-3+ـل1Del+
س	605	588	17	97.19	2.81	11	4	95.37	94.71	-11+ـى5+ـن6+ـع2+ـص1+ـت1+ـا2Del+
ش	136	135	1	99.26	0.74	0	0	99.26	99.26	+ـث1
ص	422	419	3	99.29	0.71	2	1	98.82	98.58	-2+ـس3Del+
ض	144	133	11	92.36	7.64	0	1	92.36	91.67	+ـه1+ـن2+ـم1+ـص2+ـس3+ـر2
ط	79	76	3	96.20	3.80	1	0	94.94	94.94	-1+ـل1+ـظ2Del+
ظ	48	46	2	95.83	4.17	0	0	95.83	95.83	+ـط2
ع	811	797	14	98.27	1.73	5	1	97.66	97.53	-5+ـه3+ـم2+ـف1+ـغ2+ـص1+ـح4+ـج1Del+
غ	64	55	9	85.94	14.06	0	0	85.94	85.94	+ـم1+ـل1+ـع3+ـص1+ـح3
ف	527	520	7	98.67	1.33	1	0	98.48	98.48	-1+ـي1+ـم1+ـل3+ـق1+ـد1Del+
ق	448	444	4	99.11	0.89	0	0	99.11	99.11	+ـم3+ـت1
ك	248	246	2	99.19	0.81	0	0	99.19	99.19	+ـل1+ـق1
ل	2860	2834	26	99.09	0.91	20	8	98.39	98.11	-20+ـي6+ـى1+ـن5+ـم7+ـر1+ـج3+ـا2+ـئ1Del+
م	953	919	34	96.43	3.57	39	3	92.34	92.03	3+ـن2+ـل3+ـق1+ـف1+ـط1+ـض1+ـس4+ـر1+ـد1+ـح1+ـب1 -39+ـي14+ـهDel+
ن	874	844	30	96.57	3.43	22	3	94.05	93.71	-22+ـي3+ـه5+ـم6+ـل4+ـق1+ـع2+ـس1+ـر1+ـت3+ـب4Del+
ه	966	952	14	98.55	1.45	2	0	98.34	98.34	-2+ـم3+ـل2+ـع3+ـد1+ـح2+ـج1+ـت1+ـة1Del+
و	742	733	9	98.79	1.21	2	1	98.52	98.38	-2+ـم1+ـق2+ـر4+ـت2Del+
لأ	32	32	0	100	0.00	0	0	100	100	
لا	192	192	0	100	0.00	0	0	100	100	
ى	480	444	36	92.50	7.50	0	0	92.50	92.50	+ـي28+ـن4+ـر4
ي	963	943	20	97.92	2.08	13	5	96.57	96.05	-13+ـى5+ـم1+ـل5+ـص1+ـس1+ـح1+ـت3+ـب1+ـا2Del+
Blnk	4352	4306	46	98.94	1.06	8	3	98.76	98.69	-8+ـلأ1+ـن1+ـر40+ـح1+ـا1+ـأ2Del+
Ins	56	0	56	0.00	100	0	0	0.00	0.00	-1+ـص1+ـس4+ـذ1+ـد4+ـخ4+ـح1+ـج3+ـت1+ـب9+ـا2+ـؤ1 -3+ـي5+ـو1+ـن3+ـم3+ـل8+ـع1+ـضBlnk+

Table 7-4: Results comparisons before and after Post-processing for M08-A02-C01.

	Before Post-processing					After Post-processing				
Letter	Substitution	Deletion	Insertion	Correctness	Accuracy	Substitution	Deletion	Insertion	Correctness	Accuracy
ء	0	2	0	98.18	98.18	0	2	0	98.18	98.18
آ	0	0	0	100	100	0	0	0	100	100
أ	0	2	0	99.51	99.51	0	2	0	99.51	99.51
ؤ	5	0	0	84.38	84.38	5	0	1	84.38	81.25
إ	0	0	0	100	100	0	0	0	100	100
ئ	7	0	0	90.28	90.28	0	0	0	100	100
ا	16	106	4	93.39	93.17	15	116	2	92.86	92.76
ب	28	20	9	92.73	91.36	21	20	9	93.79	92.42
ة	4	0	0	98.15	98.15	6	0	0	97.22	97.22
ت	27	18	1	89.91	89.69	17	17	1	92.39	92.17
ث	5	5	0	93.55	93.55	3	5	0	94.84	94.84
ج	9	1	3	93.01	90.91	4	1	3	96.50	94.41
ح	46	0	1	82.03	81.64	12	0	1	95.31	94.92
خ	14	0	4	84.09	79.55	3	0	4	96.59	92.05
د	10	1	4	95.24	93.51	3	1	4	98.27	96.54
ذ	3	1	1	96.85	96.06	1	1	1	98.43	97.64
ر	24	0	0	95.52	95.52	12	0	0	97.76	97.76
ز	11	2	0	79.03	79.03	1	3	0	93.44	93.44
س	20	11	3	94.88	94.38	17	11	4	95.37	94.71
ش	1	0	0	99.26	99.26	1	0	0	99.26	99.26
ص	4	2	1	98.58	98.34	3	2	1	98.82	98.58
ض	13	0	1	90.97	90.28	11	0	1	92.36	91.67
ط	4	1	0	93.67	93.67	3	1	0	94.94	94.94
ظ	3	0	0	93.75	93.75	2	0	0	95.83	95.83
ع	42	4	1	94.33	94.21	14	5	1	97.66	97.53
غ	14	0	0	78.13	78.13	9	0	0	85.94	85.94
ف	10	0	0	98.11	98.11	7	1	0	98.48	98.48
ق	5	0	0	98.88	98.88	4	0	0	99.11	99.11
ك	3	0	0	98.79	98.79	2	0	0	99.19	99.19
ل	18	31	7	98.28	98.03	26	20	8	98.39	98.11
م	49	39	5	90.77	90.24	34	39	3	92.34	92.03
ن	27	22	3	94.39	94.05	30	22	3	94.05	93.71
ه	22	1	0	97.62	97.62	14	2	0	98.34	98.34
و	10	1	2	98.52	98.25	9	2	1	98.52	98.38
لأ	0	0	0	100	100	0	0	0	100	100
لا	0	0	3	100	98.44	0	0	0	100	100
ى	45	0	0	90.63	90.63	36	0	0	92.50	92.50
ي	37	13	20	94.81	94.81	20	13	5	96.57	96.05
Blank	46	8	0	98.76	98.76	46	8	3	98.76	98.69
Total	352	122	43			270	115	26		

7.5 Summary and Conclusions

This chapter proposes techniques for the post-processing phase which aims at enhancing the recognition rate for our OCR system. Both character-level and word level post-processing are used. The character level post-processing depends on encoding the shapes of letters into their letter codes. On the other hand, the word level post-processing uses a domain dictionary to identify the incorrect words. The proposed post-processing module uses the learned knowledge from the OCR system to prioritize the correcting operations between characters. Moreover, the module is flexible and can be enhanced further to accept rule based correction. Two examples of such rules were investigated: deleting the blank, if any, at the end of line, and replacing multiple consecutive blanks by one blank.

The post-processing phase at the character level managed to improve the recognition rates for single font classifications, while improvements for the multi-font classifications were achieved using the post-processing phase at word level. The increases in recognition rates for single fonts and multi-fonts exceeded 1% and 0.8%, respectively.

The proposed post-processing techniques for Arabic OCR have several advantages. It has managed to improve the recognition rate. It does not require much processing time as it takes only seconds on X86-based PC Intel® Core™ 2 Duo CPU T8300 @ 2.40GHZ. Moreover, the results could be used by other researchers to improve their recognition rates.

Chapter 8. Conclusions and Suggestions for Future work

8.1 Introduction

This thesis presents new algorithms for efficient recognition of off-line printed Arabic text using HMM. This chapter is the conclusion of the thesis. Section 8.2 provides general conclusions. Section 8.3 gives more detailed conclusions. Section 8.4 pinpoints major contributions to the field. Possible future work is suggested in Section 8.5. The implemented algorithms along with the datasets and tools developed are provided in the enclosed CD-ROM (See Appendix A).

8.2 Overall Conclusion

Basic research in automatic printed Arabic text recognition was conducted and several related algorithms and techniques were developed. The algorithms and techniques developed were implemented to prove the validity of the research results.

Statistical and syntactical analysis for Arabic text was carried out to estimate the probabilities of occurrences of Arabic character for use with HMM and other techniques.

Since there is no adequate data for printed Arabic text recognition research that is freely available, work towards making new benchmark dataset for the research was addressed. To make the data preparation task more feasible in terms of effort and time, a new minimal set of Arabic characters to represent Arabic text was developed. The proposed script contains all basic shapes of Arabic letters. The script provides efficient representation for Arabic text in terms of effort and time. This minimal text has facilitated the generation of data for use in automatic Arabic text recognition and has reduced the effort and time required.

Based on the success of using Hidden Markov models (HMM) for speech and text recognition, the use of HMM for the automatic recognition of Arabic text was investigated. The HMM technique managed to adapt to noise and font variations. In addition, it does not require word or character segmentation of Arabic line images. The segmentation is a by-product of the recognition.

The research work behind this thesis has resulted in the improvement of the state of the art in Arabic text recognition. Practical printed Arabic data for OCR has been prepared and has been made available for researchers. New efficient feature extraction algorithms were proposed and developed. Higher recognition rates were achieved. A flexible prototype post-processing system was designed and implemented to improve Arabic OCR output for better recognition rates.

8.3 Detailed Conclusions

In this thesis the problem of automatic recognition of printed Arabic text using HMM was addressed. The emphasis was on the feature extraction and classification phases as these phases have more research potential and need with respect to automatic Arabic text recognition. Concluding remarks on this research are listed as follows:

- Analytical statistics of standard classical Arabic text of two books were pursued. The statistics were mainly on the frequencies of different shapes of Arabic alphabets and written Arabic syllables of word. One use of such statistics is to help in preparing suitable data that fairly and naturally represents classic standard

Arabic. The statistics could also be used for enhancing the recognition of Arabic OCR system. The statistics could also be used in a post-processing phase following the classification phase to correct possible mistakes. The statistics are made available for researchers.

- Since there are no adequate dataset benchmarks for printed Arabic text recognition research, work towards making new data for the research was addressed. Two datasets have been introduced and made available for researchers. The databases were prepared for eight different fonts: Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Traditional Arabic, and Andalus.
- While preparing the database a novel minimal Arabic script has been developed to ensure the coverage of all basic shapes of Arabic alphabets. The developed minimal Arabic script consists of few Arabic words that contain all basic shapes of all Arabic alphabets.
- New language-independent feature extraction schemes were proposed and used. The schemes were based on extracting a small number of single-type features. These schemes were used for automatic recognition of off-line Arabic text using HMM. The performance analysis of the HMM with different numbers of features, different sizes of sliding windows, different numbers of states and different codebook sizes were presented. The recognition technique was applied for each font of the eight Arabic fonts under study as well as several categories of multi-font groups.
- For training and testing the used techniques, the prepared two database sets of line images were used. The testing and training line images were randomly selected from the datasets.

- The experimental results indicated the effectiveness of the proposed technique in the automatic recognition of off-line printed Arabic text with different types of fonts. They showed the effectiveness of the feature extraction schemes used, which depend on a small number of simple and effective features that can be computed quickly.
- The recognition technique has been applied to eight different Arabic fonts. They all gave acceptable recognition rates. All results are new records in the recognition of printed Arabic text. For single font recognition, the accuracy percentage range was: 97.86 - 99.9. For multi-font recognitions, the accuracy percentages vary from 95.61 for the 8 fonts together, to 99.2 for a category of 2 fonts.
- The same model of Arabic text recognition without change or enhancement in training and testing has been used for English and Bangla text recognition. English was chosen to represent Latin languages and Bangla was chosen to represent Indic languages. The algorithm has been tested using the Hidden Markov Models with character accuracy of 98.46% for English, and 95.25% for Bangla. The results showed that the proposed feature extraction technique is language independent and captures enough features of the text images.
- The proposed techniques for OCR post-processing included character-level post-processing and word level post-processing. In character level post-processing encoding the shapes of letters into their letter codes was used. In word level post-processing, the incorrect words were identified through a domain dictionary. Then, trials to correct each incorrect word through single substitution, deletion, or insertion were pursued. The post-processing module used the learned knowledge from the OCR system to prioritize the correcting operations between characters.

The post-processing stage at the character level has proven to give positive improvements in recognition rates for single font classifications of up to 1%. The post-processing stage at the word level improved the multi-font classifications by up to 0.8%.

8.4 Contribution

Several contributions were evolved while developing the algorithms for optical recognition of printed Arabic text. The following subsections list the major contributions to advances of the field.

8.4.1 Providing Statistical Analysis for Standard Classical Arabic

The pursued statistical analysis of two books representing standard classical Arabic is made available for researchers. The analysis is the first of its type to include the shapes of the letters and the written syllables for classic Arabic. Partial results were published in [*140*].

8.4.2 Database Preparation for possibly being a Benchmark

The two prepared datasets of Arabic line images cover all Arabic letters and all basic shapes of the letters. The datasets are made available for researchers with the recognition rates that have been achieved [*129*]. Moreover, the testing and training sets are also provided. This will allow researchers to compare their results with the results reported here and will make these datasets become a benchmark for printed Arabic text.

8.4.3 Minimal Arabic Script

The minimal Arabic script that has been proposed could be used to build benchmark databases for handwritten Arabic text. The script consists of only three lines. This encourages many volunteers to participate with their handwritings. Moreover, as the procedures and the algorithms of finding the minimal Arabic script were stated clearly, they could be used to advise different minimal scripts in different domains. The details related to this work were reported in [*134*] [*135*].

8.4.4 New Feature Extraction Algorithms

The new feature extraction techniques provide language independent tools to select features of text for OCR. The techniques were reported in [*148*].

8.4.5 Higher Recognition for Both Single-Font and Multi-Font

The achieved recognition rates are believed to be new records in the recognition of printed Arabic text. Involving the shapes of letters instead of letters in the recognition process is believed to be new in Arabic OCR recognition. Single font recognition results were reported in [*147*].

8.4.6 Multi-Font Classification Through Categorization

A new technique to tackle the multi-font recognition problem by categorizing the fonts into categories was introduced. Such a technique was not addressed before.

8.4.7 A Flexible Prototype Post-Processing System

A flexible prototype post-processing system was designed and implemented to improve Arabic OCR output for better recognition rates.

8.5 Possible Future Work

The results in this thesis provide a strong foundation for future work in the field of Arabic OCR, both printed and handwritten. There are several lines of research arising from this work which should be pursued. These are natural extensions to the presented work. The following sections outline the main proposed lines of research in relation to the main contributions of the thesis.

8.5.1 Database Benchmarks

Expanding the benchmark databases by building a handwritten database using the proposed minimal Arabic script.

8.5.2 Minimal Arabic Script

Developing new Minimal Scripts for different languages that uses Arabic letters such as Urdu and Farsi will help the advances in OCR for those languages.

8.5.3 Handwritten recognition

The presented techniques could be pursued to recognize Arabic handwritten text. Experimenting with the suggested feature extraction schemes and fine tuning them to work with Arabic handwritten recognition is a possible future direction.

8.5.4 Feature Extraction with more languages

Using the proposed feature extraction schemes in the recognition of other languages such as Chinese and Japanese languages seems to be promising. Investigations of such issues are needed. The sign language also is a candidate for similar investigation.

8.5.5 Post-processing

Finally, one future direction is to expand the post-processing module to include more OCR learning knowledge. It could be also enhanced by adding morphology and syntax stages to it.

References

[1] Robert A. Cote, "Choosing One Dialect for the Arabic Speaking World: An Unnecessary, Unpopular and Unlikely Status Planning Dilemma," *Arizona Working Papers in SLA & Teaching*, vol. 16, pp. 75-97, 2009, http://w3.coh.arizona.edu/awp.

[2] United Nations, *Basic Facts About The United Nations*.: United Nations, 2004.

[3] United Nations. (2002, Dec.) Department for General Assembly and Conference Management. [Online]. http://www.un.org/Depts/DGACM/faq_languages.htm

[4] National Geographic. (2004, Feb.) National Geographic. [Online]. http://news.nationalgeographic.com/news/2004/02/0226_040226_language.html

[5] United Nations. (2006, Jan.) United Nations Arabic Language Programme. [Online]. http://www.un.org/depts/OHRM/sds/lcp/Arabic/

[6] Horst Bunke and Tamas Varga, "Off-line Roman Cursive Handwriting Recognition," in *Digital Document Processing: Major Directions and Recent Advances*.: Springer, 2007, vol. 20, pp. 165-173.

[7] Y. Al-Ohali, M. Cheriet, and C. Suen, "Databases for recognition of handwritten Arabic cheques," *Pattern Recognition*, vol. 2003, pp. 111-121, 2003.

[8] Jianying Hu, Sok Gek Lim, and Michael K. Brown, "Writer independent on-line handwriting recognition using an HMM approach," *Pattern Recognition*, vol. 2000, no. 33, pp. 133-147, 2000.

[9] Ophir Frieder, Gady Agam, Shlomo Argamon, and David Grossman, "Cross-Lingual Knowledge Management for Complex Arabic Documents," Department of Computer Science, Illinois Institute of Technology, 2002.

[10] Latifa Al-Sulaiti, "Designing and Developing a Corpus of Contemporary Arabic," State University of New York at Buffalo, The University of Leeds, Leeds, UK, MSc Thesis 2004.

[11] M.A. Rashwan, M.W. Fakhr, M. Attia, and M.S. El-Mahallawy, "Arabic OCR System Analogous to HMM-Based ASR Systems; Implementation and Evaluation," *Journal Of Engineering And Applied Science*, vol. 54, pp. 653-672, 2007, Faculty of Engineering Cairo University.

[12] Alessandro Vinciarelli, Samy Bengio, and Bunke Horst, "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 709-720, 2004.

[13] N. Mezghani, M. Cheriet, and A. Mitiche, "On-line recognition of handwritten Arabic characters using a Kohonen neural network," , 2002, pp. 490-495.

[14] Mary L. Manfredi, Sung-Hyuk Cha, Sungsoo Yoon, and C. Tappert, "Handwriting Copybook Style Analysis of Pseudo-Online Data," , 2005, pp. D2.1--D2.5.

[15] Ramin Halavati, Mansour Jamzad, and Mahdieh Soleymani, "A Novel Approach to Persian Online Hand Writing Recognition," *Transactions on Engineering, Computing and Technology*, vol. 6, pp. 232-236, 2005.

[16] Adnan Amin, "Off line Arabic Character Recognition - A Survey," in *4th International Conference Document Analysis and Recognition (ICDAR '97)*, 1997, pp. 596-599.

[17] Adnan Amin, "Off-line Arabic character recognition: the state of the art," *Pattern Recognition*, vol. 31, pp. 517-530, 1998.

[18] Liana Lorigo and Venu Govindaraju, "Off-line Arabic Handwriting Recognition: A survey," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 712-724, 2006.

[19] Asim Nabawi and Sabri A. Mahmoud, "Arabic Optical Text Recognition: A Classified Bibliography," *Engineering Research Bulletin, Minufiyah University, Egypt*, vol. 23, pp. 79-131, 2000.

[20] Ramzi A. Haraty and Catherine Ghaddar, "Arabic Text Recognition," *International Arab Journal of Information Technology*, vol. 1, pp. 156-163, 2004.

[21] J. Trenkle, A. Gillies, E. Erlandson, S. Schlosser, and Stan Cavin, "Advances In Arabic Text Recognition," in *Symposium on Document Image Understanding Technology (SDIUT 2001)*, Columbia, Maryland, 2001, pp. 159-168.

[22] Gheith A. Abandah and Mohammed Zeki Khedher, "Printed and Handwritten Arabic Optical Character Recognition – Initial Study," The University of Jordan, Amman, Jordan, 2004.

[23] Kareem Darwish, "Probabilistic Methods for Searching OCR-Degraded Arabic Text," Electrical and Computer Engineering, University of Maryland, University of Maryland, College Park, PhD Thesis 2003.

[24] Gregory Raymond Ball, "Arabic handwriting recognition using machine learning approaches," Computer Science and Engineering, State University of New York at Buffalo, 2007.

[25] Tim Klassen, "Towards Neural Network Recognition of Handwritten Arabic Letters," Dalhousie University, Dalhousie University, Halifax, Nova Scotia, MSc Thesis 2001.

[26] P. Burrow, "Arabic Handwriting Recognition," The University of Edinburgh, The University of Edinburgh, Edinburgh, UK, MSc Thesis 2004.

[27] Atallah AL-Shatnawi and Khairuddin Omar, "Methods of Arabic Language Baseline Detection - The State of Art," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 8, no. 10, pp. 137-143, October 2008.

[28] Abdurazzag Ali Aburas and Mohamed E. Gumah, "Arabic handwriting recognition: Challenges and solutions," , vol. 2, 2008, pp. 1-6.

[29] Mahtab Nikkhou and Khalid Choukri, "Survey on Arabic Language Resources and Tools in the Mediterranean Countries," NEMLAR, Center for Sprogteknologi, University of Copenhagen, 2005.

[30] Abdelmalek Zidouri, "ORAN: A Basis For An Arabic Ocr System," in *International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, 2004, pp. 703-706.

[31] A. Zidouri, M. Sarfraz, S. N. Nawaz, and M. J. Ahmad, "Pc Based Offline Arabic Text Recognition System," in *7th International Symposium on Signal Processing and its Applications*, Paris, France, 2003, pp. 431-434.

[32] T. Sari and M. Sellami, "Cursive Arabic Script Segmentation and Recognition System," *International Journal of Computers and Applications*, vol. 27, pp. 161-168, 2005.

[33] Muhammad Sarfraz, Syed Nazim Nawaz, and Abdulaziz Al-Khuraidly, "Offline Arabic Text Recognition system," in *International Conference on Geometric Modeling and Graphics (GMAG'03)*, London, England, 2003, pp. 30-36.

[34] Christopher LaPre, Ying Zhao, Christopher Raphael, Richard Schwartz, and John Makhoul, "Multi-Font Recognition Of Printed Arabic Using The BBN BYBLOS Speech Recognition System," in *IEEE International Conference On Acoustics, Speech And Signal Processing*, 1996, pp. 2136-2139.

[35] Latifa Hamami and Daoud Berkani, "Recognition System For Printed Multi-Font And Multi-Size Arabic Characters," *The Arabian Journal for Science and Engineering*, vol. 27, pp. 57-72, 2002.

[36] Andrew Gillies, Erik Erlandson, John Trenkle, and Steve Schlosser, "Arabic Text Recognition System," in *Symposium on Document Image Understanding Technology*, Annapolis, Maryland, 1999, pp. 234-244, Andrew Gillies (123 N. Ashley Street, Suite 120, AnnArbor, MI 48104); Erik Erl (123 N. Ashley Street, Suite120, Ann Arbor, MI 48104); John Trenkle (123 N. AshleyStreet, Suite 120, Ann Arbor, MI 48104); SteveSchlosser (123 N. Ashley Street, Suite 120, Ann Arbor,MI 48104).

[37] John Cowell and Fiaz Husain, "A Fast Recognition System for Isolated Arabic characters," in *IEEE Conference on Information Visualization 2002*, London, UK, 2002, pp. 650-654.

[38] Anthony Cheung, Mohammed Bennamoun, and Neil W. Bergmann,

"Implementation Of A Statistical Based Arabic Character Recognition System," in *IEEE TENCON - Speech and Image Technologies for Computing and Telecommunications*, 1997, pp. 531-534.

[39] A. Cheung, M. Bennamoun, and N. W. Bergmann, "A Recognition-Based Arabic Optical Character Recognition System," in *IEEE International Conference on Systems, Man, and Cybernetics*, 1998, pp. 4189-4194.

[40] A. Cheung, M. Bennamoun, and N. W. Bergmann, "An Arabic optical character recognition system using recognition-based segmentation," *Pattern Recognition*, vol. 34, pp. 215-233, 2001.

[41] Abdurazzag Ali Aburas and Salem M. Rehiel, "Off-line Omni-style Handwriting Arabic Character Recognition System Based on Wavelet Compression," *Arab Research Institute For Science \& Engineering*, vol. 3, pp. 123-135, 2007.

[42] Mustafa Syiam, T. M. Nazmy, A. E. Fahmy, H. Fathi, and K. Ali, "Histogram clustering and hybrid classifier for handwritten Arabic characters recognition," in *The 24th IASTED international conference on Signal processing, pattern recognition, and applications*, 2006, pp. 44-49.

[43] Mehdi Dehghan, Karim Faez, Majid Ahmadi, and Malayappan Shridhar, "Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM," *Pattern Recognition*, vol. 2001, pp. 1057-1065, 2001.

[44] H. Bentouns and M. Batouche, "Incremental support vector machines for handwritten Arabic character recognition," in *Information and Communication Technologies: From Theory to Applications*, 2004, pp. 477-478.

[45] Nadia Ben Amor and Najoua Essoukri Ben Amra, "Multifont Arabic Character Recognition Using Hough Transform and Hidden Markov Models," in *4th International Symposium on Image and Signal Processing and Analysis*, 2005, pp. 285-288.

[46] Issam Bazzi, Chris LaPre, John Makhoul, Chris Raphael, and Richard Schwartz, "Omnifont and Unlimited-Vocabulary OCR for English and Arabic," in *4th International Conference Document Analysis and Recognition (ICDAR '97)*, 1997, pp. 842-846.

[47] Nadir Farah, Labiba Souici, and Mokhtar Sellami, "Classifiers combination and syntax analysis for Arabic literal amount recognition," *Engineering Applications of Artificial Intelligence*, vol. 2006, pp. 29-39, 2006.

[48] http://www.ifnenit.com/, IFN/ENIT-database – Database Of Handwritten Arabic Words, 2006.

[49] Mario Pechwitz and Volker Märgner, "HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT- Database," in *The Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, 2003, pp. 890-894.

[50] Mario Pechwitz, Samia Snoussi Maddouri, Volker Märgner, Noureddine Ellouze, and Hamid Amiri, "IFN/ENIT - Database Of Handwritten Arabic Words," in *CIFED*, 2002, pp. 129-136.

[51] V. Märgner, M. Pechwitz, and H. El Abed, "ICDAR 2005 Arabic Handwriting Recognition Competition," in *International Conference on Document Analysis and Recognition*, 2005, pp. 70-74.

[52] S. Al-Ma'adeed, C. Higgens, and D. Elliman, "Off-line recognition of handwritten Arabic words using multiple hidden Markov models," *Knowledge-Based Systems*, vol. 2004, pp. 75-79, 2004.

[53] Somaya Al-Ma'adeed, Dave Elliman, and Colin Higgins, "A Data Base for Arabic Handwritten Text Recognition Research," *International Arab Journal on Information Technology*, vol. 1, pp. -, 2004.

[54] S. Al-Ma'adeed, D. Elliman, and C. Higgins, "A Data Base for Arabic Handwritten Text Recognition Research," in *The Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 485-489.

[55] Y. A. Alotaibi, "High Performance Arabic Digits Recognizer Using Neural Networks," in *The International Joint Conference On Neural Networks*, 2003, pp. 670-674.

[56] Mahdieh Soleymani and Farbod Razzazi, "An Efficient Front-End system for Isolated Persian/Arabic Character Recognition of Handwritten Data-Entry Forms," in *WSEAS Multiconference*, 2003, p. 6.

[57] Yousef Al-Ohali, Mohamed Cheriet, and Ching Suen, "Databases For Recognition Of Handwritten Arabic Cheques," in *Seventh International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, 2000, pp. 601-606.

[58] S. Snoussi Maddouri, H. Amiri, A. Belaïd, and Ch. Choisy, "Combination of Local and Global Vision Modelling for Arabic Handwritten Words Recognition," in *Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 128-135.

[59] V. Märgner and M. Pechwitz, "Synthetic Data for Arabic OCR System Development," in *The 6th International Conference on Document Analysis and Recognition, ICDAR'01*, 2001, pp. 1159-1163.

[60] Alaa Hamid and Ramzi Haraty, "A Neuro-Heuristic Approach for Segmenting Handwritten Arabic Text," in *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'01)*, Beirut, Lebanon, 2001, pp. 110-113.

[61] Nawwaf Kharma, Maher Ahmed, and Rabab Ward, "A New Comprehensive Database of Handwritten Arabic Words, Numbers, and Signatures used for OCR Testing," in *1999 IEEE Canadian Conference on Electrical and Computer Engineering*, 1999, pp. 766-768.

[62] Nayer Wanas, Mohamed S. Kamel, Gasser Auda, and Fakhreddine Karray, "Feature-based decision aggregation in modular neural network classifiers," *Pattern Recognition Letters*, vol. 1999, pp. 1353-1359, 1999.

[63] John Makhoul, Richard Schwartz, Christopher Lapre, and Issam Bazzi, "A Script-Independent Methodology For Optical Character Recognition," *Pattern Recognition*, vol. 31, pp. 1285-1294, 1998.

[64] J. Trenkle, E. Erlandson, A. Gillies, and S. Schlosser, "Arabic Character Recognition," in *Symposium on Document Image*, Bowie, Maryland, 1995, pp. 191-195.

[65] M. Melhi, "Off-line Arabic Cursive Handwriting Recognition Using Artificial Neural Networks," University of Bradford, Bradford University, Bradford, UK, PhD Thesis 2001.

[66] Najoua Essoukri Ben Amara, Omar Mazhoud, Noura Bouzrara, and Noureddine Ellouze, "ARABASE: A Relational Database for Arabic OCR Systems," *Int. Arab J. Inf. Technol*, vol. 2, pp. 259-266, 2005.

[67] Toufik Sari, Mokhtar Sellami, and Labiba Souici, "Off-line Handwritten Arabic Character Segmentation Algorithm: ACSA," in *The Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR’02)*, 2002, pp. 452-457.

[68] S. N. Nawaz, M. Sarfraz, A. Zidouri, and W. G. Al-Khatib, "An Approach To Offline Arabic Character Recognition Using Neural Networks," in *10th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2003)*, vol. 3, 2003, pp. 1328-1331.

[69] M. Pechwitz and V. Märgner, "Baseline Estimation For Arabic Handwritten Words," in *Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 479-484.

[70] Mohammad S. Khorsheed and William F. Clocksin, "Structural Features of Cursive Arabic Script," in *10th British Machine Vision Conference*, vol. 2, 1999, pp. 422-431.

[71] Wasfi Al-Khatib and Sabri Mahamud, "Toward Content-Based Indexing and Retrieval of Arabic Manuscripts," KIng Fahd University of Petroleum \& Minerals, 2006.

[72] Muhammad Sarfraz and S. A. Shahab, "An Efficient Scheme for Tilt Correction in Arabic OCR System," in *International Conference on Computer Graphics, Imaging and Vision: New Trends*, 2005, pp. 379-384.

[73] Sofien Touj, Najoua Essoukri Ben, and Hamid Amiri, "Generalized Hough Transform for Arabic Optical Character Recognition," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2003, pp. 1242-1246.

[74] Sofien Touj, Najoua Ben Amara, and Hamid Amiri, "Global Feature Extraction of Off-Line Arabic Handwriting," in *IEEE International Conference on Systems, Man and Cybernetics, 2002*, vol. 4, 2002, p. 4.

[75] Mohamed Fakir and M. M. Hassani, "On The Recognition Of Arabic Characters Using Hough Transform Technique," *Malaysian Journal of Computer Science*, vol. 13, pp. 39-47, 2000.

[76] Sabri Mahmoud, "Arabic Character-Recognition Using Fourier Descriptors and Character Contour Encoding," *Pattern Recognition*, vol. 27, pp. 815-824, 1994.

[77] A. Zahour, B. Taconet, P. Mercy, and S. Ramdane, "Arabic Hand-Written Text-Line Extraction," in *Sixth International Conference on Document Analysis and Recognition (ICDAR '01)*, 2001, pp. 281-285.

[78] Majid Altuwaijri and Magdy Bayoumi, "A Thinning Algorithm for Arabic Characters Using ART2 Neural Network," *IEEE Transactions On Circuits And Systems-Ii: Analog And Digital Signal Processing*, vol. 45, pp. 260-264, 1998.

[79] M. H. Shirali-Shahreza and S. Shirali-Shahreza, "Persian/Arabic Text Font Estimation using Dots," in *Sixth IEEE International Symposium on Signal Processing and Information Technology*, 2006, pp. 420-425.

[80] John Cowell and Fiaz Hussain, "Thinning Arabic Characters for Feature Extraction," in *Fifth International Conference on Information Visualization (IV'01)*, 2001, pp. 181-185.

[81] A. Zidouri, M. Sarfraz, S. A. Shahab, and S. M. Jafri, "Adaptive Dissection Based Subword Segmentation Of Printed Arabic Text," in *Ninth International Conference on Information Visualization (IV'05)*, vol. 00, 2005, pp. 239-243.

[82] Liying Zheng, Abbas Hassin, and Xianglong Tang, "A new algorithm for machine printed Arabic character segmentation," *Pattern Recognition Letters*, vol. 25, pp. 1723-1729, 2004.

[83] K. Romeo-Pakker, A. Ameur, C. Olivier, and Y. Lecourtier, "Structural-Analysis of Arabic Handwriting: Segmentation and Recognition," *Machine Vision and Applications*, vol. 8, pp. 232-240, 1995.

[84] G. Olivier, H. Miled, K. Romeo, and Y. Lecourtier, "Segmentation and coding of Arabic handwritten words," in *13th International Conference on Pattern Recognition*, vol. 3, 1996, pp. 264-268.

[85] Mohamed F. Tolba, Gamal Abdul Moty, and Ahmed Mahmoud, "Segmentation Free Approach for Printed Arabic Text Recognition," *International Journal of Computers and Their Applications*, vol. 10, pp. 94-102, 2003.

[86] M. S. Khorsheed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)," *Pattern Recognition Letters*, vol. 28, pp. 1563-1571, 2007.

[87] S. Al-Ma'adeed, C. Higgins, and D. Elliman, "Recognition of Off-Line

Handwritten Arabic Words Using Hidden Markov Model Approach," in *International Conference on Pattern Recognition (ICPR 2002),* 2002, pp. 481-484.

[88] Deya Motawa, Adnan Amin, and Robert Sabourin, "Segmentation of Arabic Cursive Script," in *4th International Conference Document Analysis and Recognition (ICDAR '97)*, vol. 2, 1997, pp. 625-628.

[89] Liana Lorigo and Venu Govindaraju, "Segmentation and Pre-Recognition of Arabic Handwriting," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, 2005, pp. 605-609.

[90] Ahmed M. Elgammal and Mohamed A. Ismail, "A Graph-Based Segmentation and Feature Extraction Framework for Arabic Text Recognition," in *The 6th International Conference on Document Analysis and Recognition (ICDAR01)*, 2001, pp. 622-626.

[91] Ahmed Kandil and Ahmed El-Bialy, "Arabic OCR: A Centerline Independent Segmentation Technique," in *The 2004 International Conference on Electrical, Electronic, and Computer Engineering (ICEEC '04)*, 2004, pp. 412-415.

[92] Karim Hadjar and Rolf Ingold, "Arabic Newspaper Page Segmentation," in *Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, vol. 2, 2003, pp. 895-899.

[93] A.M. Gouda and M.A. Rashwan, "Segmentation of connected Arabic characters using hidden Markov models," in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA2004)*, 2004, pp. 115-119.

[94] A. Broumandnia, M.J. Shanbehzadeh, and M. Nourani, "Segmentation of Printed Farsi/Arabic Words," in *IEEE/ACS International Conference on Computer Systems and Applications*, 2007, pp. 761-766.

[95] M. Cheriet, N. Kharma, C. L. Liu, and C. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*.: Wiley-Interscience, 2007.

[96] N. Iwayama and K. Ishigaki, "Adaptive Context Processing in On-Line Handwritten Character Recognition," , 2000, pp. 469-474.

[97] Afshin Ebrahimi and Ehsanollah Kabir, "A pictorial dictionary for printed Farsi subwords," *Pattern Recognition Letters*, vol. 29, pp. 656-663, 2008.

[98] Ahmad T. Al-Taani, "An Efficient Feature Extraction Algorithm for the Recognition of Handwritten Arabic Digits," *International Journal of Computational Intelligence*, vol. 2, pp. 107-111, 2005.

[99] Sabri Mahmoud, "Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models," *Signal Processing*, vol. 88, pp. 844-857, 2008.

[100] Angshul Majumdar, "Bangla Basic Character Recognition Using Digital Curvelet Transform," *Journal of Pattern Recognition Research*, vol. 2, pp. 17-26, 2007.

[101] Sargur N. Srihari, Catalin I. Tomai, Bin Zhang, and Sangjik Lee, "Individuality of Numerals," in *The Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, Washington, DC, USA, 2003, p. 1096.

[102] Christian Gagne and Marc Parizeau, "Genetic Engineering of Hierarchical Fuzzy Regional Representations for Handwritten Character Recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 8, pp. 223-231, 2006.

[103] Quan-Sen Sun, Sheng-Gen Zeng, Yan Liu, Pheng-Ann Heng, and De-Shen Xia, "A new method of feature fusion andits application in image recognition," *Pattern Recognition*, vol. 2005, pp. 2437-2448, 2005.

[104] Anil K. Jain, Robert P.W. Duin, and Jianchang Mao, "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4-37, 2000.

[105] Amar Gupta, M. V. Nagendraprasad, A. Liu, P. S. P., and S. Ayyadurai, "An Integrated Architecture For Recognition Of Totally Unconstrained Handwritten Numerals," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 7, pp. 753-773, 1993.

[106] R. Al-Alawi, "A Neural Network Recognition System for Isolated Handwritten Arabic Characters," , 2003, pp. 262-265.

[107] Faruq Al-Omaria and Omar Al-Jarrah, "Handwritten Indian numerals recognition system using probabilistic neural networks," *Advanced Engineering Informatics*, vol. 2004, no. 18, pp. 9-16, 2004.

[108] Peeta Basa Pat and A.G. Ramakrishnan, "Word level multi-script identification," *Pattern Recognition Letters*, vol. 29 (2008), pp. 1218-1229, 2008.

[109] Nayer M. Wanas, Mahmoud R. El-Sakka, and Mohamed S. Kamel, "Multiple Classifier Hierarchical Architecture for Handwritten Arabic Character Recognition," , 1999, pp. 2834-2838.

[110] Yi Chang, Datong Chen, Ying Zhang, and Jie Yang, "An image-based automatic Arabic translation system," *Pattern Recognition*, vol. 42, pp. 2127-2134, 2009.

[111] Ilya Zavorin and Mark Borovikov, Eugene, "Initial Results in Offline Arabic Handwriting Recognition Using Large-Scale Geometric Features," in *Symposium on Document Image Understanding Technology* , Maryland, 2005, pp. 79-88.

[112] Sameh Touj, Najoua Ben Amara, and Hamid Amiri, "Arabic Handwritten Words Recognition Based on a Planar Hidden Markov Model," *Int. Arab J. Inf. Technol*, vol. 2, pp. 318-325, 2005.

[113] Housem Miled and Najoua Essoukri Ben Amara, "Planar Markov Modeling for

Arabic Writing Recognition : Advancement State," in *Sixth Int'l Conf. Document Analysis and Recognition*, 2001, pp. 69-73.

[114] M. S. Khorsheed, "Recognizing handwritten Arabic manuscripts using a single hidden Markov model," *Pattern Recognition Letters*, vol. 2003, pp. 2235-2243, 2003.

[115] T. Sari and M. Sellami, "Morpho-Lexical analysis for correcting OCR-generated arabic words (MOLEX)," in *Frontiers in Handwriting Recognition*, 2002, pp. 461-466.

[116] S. Kanoun, A. Ennaji, Y. Lecourtier, and A. M. Alimi, "Linguistic integration information in the AABATAS arabic text analysis system," in *Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 389-394.

[117] E. Borovikov, I. Zavorin, and M. Turner, "A filter based post-OCR accuracy boost system," in *1st ACM workshop on Hardcopy document processing*, 2004, pp. 23-28.

[118] Mohammed Zeki Khedher and Gheith Abandah, "Arabic Character Recognition using Approximate Stroke Sequence," in *Third International Conference on Language Resources and Evaluation (LREC2002)*, 2002.

[119] Yousef Salem Elarian, "A Lexicon of Connected Components for Arabic Optical Text Recognition," Jordan University of Science and Technology, Jordan University of Science and Technology, Amman, Jordan, MSc Thesis 2006.

[120] I. Bazzi, R. Schwartz, and J. Makhoul, "An Omnifont Open-Vocabulary OCR System for English and Arabic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 495-504, 1999.

[121] Richard Schwartz, Chris LaPre, John Makhoul, Chris Raphael, and Ying Zhao, "Language-Independent OCR Using a Continuous Speech Recognition System," in *13th International Conference on Pattern Recognition*, vol. 3, 1996, pp. 99-103.

[122] IRIS. (2009, Nov.) I.R.I.S. - OCR software and Document Management solutions. [Online]. http://www.irislink.com/

[123] NovoDynamics. (2009, Nov.) NovoDynamics VERUS™ Standard. [Online]. http://www.novodynamics.com/

[124] Sakhr. (2009, Nov.) Sakhr Software. [Online]. http://www.sakhr.com/

[125] Nuance-Communications. (2009, Nov.) OmniPage OCR Software. [Online]. http://www.nuance.com/omnipage/

[126] Gregory A. Marton, Osama Bulbul, and Tapas Kanungo, "Performance Evaluation of Two Arabic OCR Products," in *AIPR Workshop on Advances in Computer Assisted Recognition, SPIE vol. 3584*, 1998, pp. 76-83.

[127] Gregory A. Marton, Osama Bulbul, and Tapas Kanungo, "OmniPage vs. Sakhr: Paired Model Evaluation of Two Arabic OCR Products," in *SPIE Conference on Document Recognition and Retrieval VI*, San Jose, CA, 1999, Gregory A. Marton (Center for Automation Research,University of Maryland; College Park, MD 20742); OsamaBulbul (Center for Automation Research, University ofMaryland; College Park, MD 20742); Tapas Kanungo(Center for Automation Research, University ofMaryland; College Park, MD 20742).

[128] V. Märgner, H. El Abed, and Pechwitz M., "Offline Handwritten Arabic Word Recognition Using HMM - a Character Based Approach without Explicit Segmentation," in *The 9th Colloque International Francophone sur l'Ecrit et le Document , CIFED 2006*, Fribourg, Swiss, 2006.

[129] Husni Al-Muhtaseb. (2009, Dec.) Arabic OCR. [Online]. http://faculty.kfupm.edu.sa/ics/muhtaseb/ArabicOCR

[130] Mohammad Al-Bukhari, *Al-Jame' Al-Saheeh (Sahih Al-Bukhari)*. Beirut: Dar Al-Jaleel, 2005, In Arabic.

[131] Muslem Al-Naysabouri, *Al-Jame' Al-Saheeh (Sahih Muslim)*. Beirut: Dar Al-Jaleel, 2006, In Arabic.

[132] Mohammad Al-Bukhari, *Al-Jame' Al-Saheeh (Sahih Al-Bukhari)*. Beirut: Dar Al-Jaleel, 2005, (in Arabic).

[133] Muslem Al-Naysabouri, *Al-Jame' Al-Saheeh (Sahih Muslim)*. Beirut: Dar Al-Jaleel, 2006, (in Arabic).

[134] Husni A. Al-Muhtaseb, Sabri A. Mahmoud, and Rami S. Qahwaji, "A Novel Minimal Arabic Script for Preparing Databases and Benchmarks for Arabic Text Recognition Research," in *8th WSEAS International Conference on SIGNAL PROCESSING (SIP '09)*, Istanbul, Turkey, 2009, pp. 37-43.

[135] Husni A. Al-Muhtaseb, Sabri A. Mahmoud, and Rami S. Qahwaji, "A Novel Minimal Script for Arabic Text Recognition Databases and Benchmarks," *International Journal of Circuits, Systems and Signal Processing*, pp. 145-153, 2009.

[136] Al-Fayrouzabadi, *Al Qamoos Al Muheet (in Arabic)*. Beirut: Darul Kutubul Ilmiyyah, 1996, a dictionary, (in Arabic).

[137] Abi Al-Hussein Ahmad Bin Faris Ibn Zakariyya, *Mu'jam Maqayees Al-Lughah*. Beirut: Dar Al-Jeel, 1999, (in Arabic).

[138] Al-Raghib Al-Asfahani, *Mu'jam Alfath Al-Qurani*. Darul Kutubul Ilmiyyah: Beirut, 1997, (in Arabic).

[139] Al Muhaddith. (2009, Nov.) Al Muhaddith. [Online]. http://www.muhaddith.org/

[140] Husni A. Al-Muhtaseb, Sabri Mahmoud, and Rami S. Qahwaji, "Statistical

Analysis for the support of Arabic Text Recognition," in *International Symposium on Computer and Arabic Language*, Riyadh, Saudi Arabia, 2007, (in Arabic).

[141] Mudit Agrawal and David Doermann, "Re-targetable OCR with Intelligent Character Segmentation," in *The Eighth IAPR International Workshop on Document Analysis Systems*, Nara, Japan, 2008, pp. 183-190.

[142] S. Alma'adeed, C. Higgens, and D. Elliman, "Recognition of off-line handwritten Arabic words using hidden Markov model approach," , vol. 3, 2002, pp. 481-484.

[143] H. Hassin, Abbas, Xiang-Long Tang, Jia-Feng Liu, and Wei Zhao, "Printed arabic character recognition using HMM," *Journal of Computer Science \& Technology*, vol. 19, pp. 538-543, 2004.

[144] Magdi Mohamed and Paul Gader, "Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 548-554, 1996.

[145] R. Al-Hajj, C. Mokbel, and L. Likforman-Sulem, "Combination of HMM-Based Classifiers for the Recognition of Arabic Handwritten Words," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 959-963.

[146] Razvan Bunescu and Raymond Mooney, "Statistical Relational Learning for Natural Language Information Extraction," in *Introduction to Statistical Relational Learning*, Lise Getoor and Ben Taskar, Eds.: MIT Press, 2007, ch. 19, pp. 535-552.

[147] Husni Al-Muhtaseb, Sabri Mahmoud, and Rami Qahwaji, "Recognition of Off-line printed Arabic text Using Hidden Markov Models," *Signal Processing*, vol. 88, no. 12, pp. 2902-2912, December 2008.

[148] Husni Al-Muhtaseb and Rami Qahwaji, "A Single Feature Extraction Algorithm for Text Recognition of Different Families of Languages," in *Third Mosharaka International Conference on Communications, Computers and Applications (MIC-CCA2009)*, Amman, Jordan, 2009.

[149] S. Young et al., *The HTK Book*. Cambridge, UK: Cambridge University Engineering Department, 2006, (for HTK version 3.4).

[150] Yaxin Zhang, Roberto Togneri, and Michael Alder, "Phoneme-Based Vector Quantization in a Discrete HMM Speech Recognizer," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 26-32, 1997.

[151] S. Al-Ma’adeed, "Recognition of Off-line Handwritten Arabic Words," The University of Nottingham, Nottingham, UK, PhD Thesis 2004.

[152] Mohamed El-Mahallawy, "A Large Scale HMM-Based Omni Font-Written OCR

System for Cursive Scripts," Faculty of Engineering, Cairo University, Cairo, Egypt, PhD Thesis 2008.

[153] Echeat Webmaster. (2007) echeat. [Online]. http://www.echeat.com/

[154] Abulkalam Azad Anwarullah and Akhtaruzaman M. Sulaiman, *Good manners and Islamic Culture*. Riyadh, Saudi Arabia: IslamHouse, 2008, (in Bangla http://www.islamhouse.com/tp/116951).

[155] Chong Long et al., "An Efficient Post-processing Approach for Off-Line Handwritten Chinese Address Recognition," in *the 8th International Conference on Signal Processing*, vol. 2, Guilin, China, 2006, pp. 1063-1066.

[156] O. Kolak and P. Resnik, "OCR Post-Processing for Low Density Languages," in *The Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, 2005.

[157] H. El Abed and V. Märgner, "Arabic Text Recognition Systems - State of the Art and Future Trends," in *International Conference Innovations in Information Technology (IIT 2008)*, Al-Ain, Qatar, 2008, pp. 692-696.

Appendix A. Contents of enclosed CD-ROM

The CD-ROM attached to this thesis contains useful resources related to the addressed research work. The following is an index of the attached CD-ROM:

Folder	Contents
Stats	Statistical analysis of Arabic text.
Minim	The source code and the utility to search huge corpora of Arabic script to find a set of minimum number of meaningful words that cover all Arabic alphabet-shapes. The corpora used are also included.
Bench	Datasets along with their ground truth information. This folder also includes the source code of the coding/decoding program.
Chars	Images of Arabic characters.
Class	Training and testing sets.
Raw	Raw confusion matrices and detailed analysis.
Features	Matlab code for extracting 30 features to be used with HTK. A code for normalization is also included.