

End-Shape Analysis for Automatic Segmentation of Arabic Handwritten Texts

Amani Tariq Jamal

A Thesis
In The Department of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy in Computer Science
Concordia University
Montreal, Quebec, Canada

© Amani T. Jamal, 2015.

**CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: Amani Jamal

Entitled: End-Shape Analysis for Automatic Segmentation of Arabic Handwritten
Texts

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of the University and meets the accepted standards with
respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. C. Wang

_____ External Examiner
Dr. M. Ahmadi

_____ External to Program
Dr. N. Kharma

_____ Examiner
Dr. R. Witte

_____ Examiner
Dr. L. Lam

_____ Thesis Supervisor
Dr. C. Y. Suen

Approved by: _____
Dr. V. Haarslev , Graduate Program Director

July 30, 2015 _____
Dr. A. Asif, Dean
Faculty of Engineering and Computer Science

Abstract

End-Shape Analysis for Automatic Segmentation of Arabic Handwritten Texts

Amani Tariq Jamal, Ph.D.

Concordia University, 2015

Word segmentation is an important task for many methods that are related to document understanding especially word spotting and word recognition. Several approaches of word segmentation have been proposed for Latin-based languages while a few of them have been introduced for Arabic texts. The fact that Arabic writing is cursive by nature and unconstrained with no clear boundaries between the words makes the processing of Arabic handwritten text a more challenging problem.

In this thesis, the design and implementation of an End-Shape Letter (ESL) based segmentation system for Arabic handwritten text is presented. This incorporates four novel aspects: (i) removal of secondary components, (ii) baseline estimation, (iii) ESL recognition, and (iv) the creation of a new off-line CENPARMI ESL database.

Arabic texts include small connected components, also called secondary components. Removing these components can improve the performance of several systems such as baseline estimation. Thus, a robust method to remove secondary components that takes into consideration the challenges in the Arabic handwriting is introduced. The methods reconstruct the image based on some criteria. The results of this method were subsequently compared with those of two other methods that used the same database. The results show that the proposed method is effective.

Baseline estimation is a challenging task for Arabic texts since it includes ligature, overlapping, and secondary components. Therefore, we propose a learning-based approach that

addresses these challenges. Our method analyzes the image and extracts baseline dependent features. Then, the baseline is estimated using a classifier.

Algorithms dealing with text segmentation usually analyze the gaps between connected components. These algorithms are based on metric calculation, finding threshold, and/or gap classification. We use two well-known metrics: bounding box and convex hull to test metric-based method on Arabic handwritten texts, and to include this technique in our approach. To determine the threshold, an unsupervised learning approach, known as the Gaussian Mixture Model, is used. Our ESL-based segmentation approach extracts the final letter of a word using rule-based technique and recognizes these letters using the implemented ESL classifier.

To demonstrate the benefit of text segmentation, a holistic word spotting system is implemented. For this system, a word recognition system is implemented. A series of experiments with different sets of features are conducted. The system shows promising results.

Thesis Supervisor: Ching Y. Suen

Title: Professor

To my parents

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Professor Ching Y. Suen for his persistent guidance and scientific support during my Ph.D. studies and research. Also, I would like to thank the examination committee for their comments and feedback. Many thanks to CENPARMI's research manager, Mr. Nicola Nobile, for his constructive discussion, inspirational instructions and technical assistance. Special thanks to my colleague, Malik Waqas Sagheer, for sharing his knowledge and providing me with immeasurable support. I am grateful to Dr. Chun Lei He, Dr. Muna Al-Khayat, and Jehan Janbi for their ideas and advices.

I want to thank Professor Mohamed Cheriet, Professor David Doerman, Dr. Haikal El Abed, and Dr. Neamat El-Gayar for their inspiring comments and wise advice. My gratitude goes to Dr. Andreas Fischer for arranging handwritten document analysis meetings. I am thankful to our office assistant, Ms. Marleah Blom, for administrative help, which I highly appreciate.

Most importantly, my gratitude goes to my parents who instilled the importance of education while always providing me with love and encouragements. This Ph.D. thesis and project would not be completed without their support. I want to thank my husband, Dr. Talal Basha, my sons, Mohammed and Adnan, along with my daughter, Nawal for their understandings, love, and support during my studies since January 2010. I would like to thank my parents-in-law, grandparents, siblings, relatives and friends for their prayers and immense caring.

Many thanks to Saudi Cultural Bureau and King Abdulaziz University for sponsoring my study and research. Thanks to Dr. Zainab Gabbani and Ms. Soha Mansour for their help and understanding.

Contents

List of Figures	x
List of Tables	xii
List of Abbreviations	xiii
1 Introduction.....	1
1.1 Definitions	2
1.2 Problem Statement.....	3
1.3 Motivation	4
1.4 Arabic Characteristics.....	5
1.5 Latin vs. Arabic	10
1.6 Challenges	11
1.7 Objectives	14
1.8 Proposed Method.....	15
1.8.1 Utilizing the Knowledge of Arabic Writing	15
1.8.2 Our Overall Methodology.....	16
1.9 Contributions	18
1.10 Database.....	18
1.11 Thesis Outlines	19
2 Removal of Secondary Components Using Morphological Reconstruction	21
2.1 Introduction	21
2.2 Related Work.....	22
2.3 Proposed Method.....	23
2.4 Experimental Result	24
2.5 Conclusions	26
3 Learning-based Baseline Estimation.....	28
3.1 Baseline Definition	28
3.2 Motivation	29
3.3 Baseline Error Measurement	30
3.4 Challenges	30
3.5 Related Works	31

3.6 Proposed Method	37
3.6.1 Our Method.....	38
3.6.2 Baseline Ground Truth.....	41
3.6.3 Baseline Database Generation	41
3.7 Experimental Results.....	42
3.8 Conclusions	46
4 Metric-based Segmentation	47
4.1 Introduction	47
4.2 Previous Work	48
4.2.1 Metric-based Segmentation Approaches	48
4.2.2 Classifier-based Approaches.....	49
4.2.3 Historical Documents.....	50
4.2.4 Word Segmentation Contests.....	50
4.3 Our Method.....	53
4.3.1 Distance Computation.....	53
4.3.2 Gap Classification.....	56
4.4 Experiments	58
4.5 Conclusions	60
5 Isolated Character Recognition System	61
5.1 Previous Works.....	63
5.2 Preprocessing.....	65
5.3 Feature Extraction.....	67
5.4 Recognition.....	70
5.5 Database.....	71
5.6 Experiments	71
5.7 Conclusions	74
6 CENPARMI Arabic Database for Handwriting Recognition	75
6.1 Related Works	76
6.2 Data Collection	76
6.3 Data Extraction	79
6.4 Database Overview	79
6.5 Ground Truth	80
6.6 Conclusions	81

7 End Shape Letter Recognition-Based Segmentation	82
7.1 Related Works	83
7.2 ESL-Based Segmentation	84
7.3 Our Proposed Algorithm for Text Segmentation	85
7.4 Experiments	90
7.5 Error analysis	93
7.6 Time Complexity	95
7.7 Comparison of Results with Arabic Text Segmentation	95
7.8 Conclusions	96
8 Arabic Handwritten Word Recognition	97
8.1 Related Works	98
8.2 Database	99
8.3 Word Recognition System	99
8.3.1 Preprocessing	99
8.3.2 Feature Extraction	99
8.3.3 Recognition	100
8.4 Experiments and Results	100
8.5 Comparison with Arabic word recognition system	102
8.6 Conclusions	102
9 Impact of Text Segmentation on Word Spotting	103
9.1 Word Spotting in the Arabic Language	104
9.2 Performance Evaluation	105
9.3 Method Implemented	105
9.4 Experimentation	106
9.5 Conclusions	108
10 Conclusions and Future Works	109
10.1 Concluding Remarks	109
10.2 Future Works	110
References	113

List of Figures

Figure 1: Arabic character shapes in different positions	6
Figure 2: Hamza's positions	7
Figure 3: An Arabic word with three different NLC letters	7
Figure 4: Arabic words with different numbers of PAWs	7
Figure 5: Main and secondary components of an Arabic word	8
Figure 6: Some representation of dots	8
Figure 7: A baseline of an Arabic word	9
Figure 8: Arabic directional markings	9
Figure 9: Same Arabic word with different directional markings	9
Figure 10: Some Arabic calligraphic styles	10
Figure 11: Types of recognition systems	13
Figure 12: Intra word gaps	13
Figure 13: Printed (below) and handwritten (above) Arabic texts	13
Figure 14: Intra and inter-word gaps in Arabic texts	14
Figure 15: Letter Noon in different positions	15
Figure 16: Overview of our methodology	16
Figure 17: Overall methodology	17
Figure 18: Main and secondary components with almost similar dimensions	22
Figure 19: Result of our secondary component removal method	25
Figure 20: Error analysis	26
Figure 21: Main properties of baseline	29
Figure 22: Baseline estimation challenges	32
Figure 23: Proposed method	38
Figure 24: Baseline range	39
Figure 25: Horizontal projection of text image	40
Figure 26: Centroids of convex hulls	40
Figure 27: Line segments using Hough Transform	41
Figure 28: Distribution of the images with respect to the number of words	42
Figure 29: Error of baseline estimation	45
Figure 30: Steps illustration of metric-based segmentation using BBs	54
Figure 31: Steps illustration of metric-based segmentation using CHs	55
Figure 32: Steps illustration of metric-based segmentation using Baseline Dependent distance	56
Figure 33: Isolated character recognition system	62
Figure 34: Noise removal	65
Figure 35: Binarization process	66
Figure 36: White space removal	66
Figure 37: Skeletonization	67
Figure 38: Robert's Filter Mask for Extracting Gradient Features	68
Figure 39: Gradient features	68
Figure 40: Confusion between classes	73

Figure 41: Filled form.....	78
Figure 42: Some samples of endWord class letters	85
Figure 43: Block diagram of the proposed method	86
Figure 44: Steps of text segmentation algorithm	91
Figure 45: Some common sources of errors	94
Figure 46: Sobel masks for extracting gradient features	100

List of Tables

Table 1: Arabic vs. English.....	11
Table 2: Statistics of number of images with two and three words in IFN/ENIT database	19
Table 3: Comparison of secondary components removal methods	26
Table 4: Results of some methods reported in the literature	36
Table 5: Challenges related to baseline methods.....	37
Table 6: Baseline estimation results of the first experiment.....	43
Table 7: Result of 74 classes.....	44
Table 8: Results of learning-based baseline estimation.....	44
Table 9: Comparison among baseline estimation methods.....	45
Table 10: Results of the participated methods in segmentation contests in Latin scripts.....	52
Table 11: BB results.....	59
Table 12: CH results	59
Table 13: Baseline dependent result	60
Table 14: Results of our method using all the classes of Arabic characters.....	72
Table 15: Comparison between different methods	72
Table 16: Combined classes.....	73
Table 17: Experimental results with different features.....	74
Table 18: Letter shapes	77
Table 19: Statistics of letter shapes.....	80
Table 20: Handwritten Indian Digits	80
Table 21: Example of the ground truth data for Arabic letter shape dataset	81
Table 22: Results of Arabic word segmentation method.....	84
Table 23: Results of metric-based and ESL-based methods using subsets of IFN/ENIT	92
Table 24: Results of our algorithm of text segmentation algorithm	93
Table 25: Time complexity of the systems	95
Table 26: Comparisons of methods of Arabic handwritten text segmentation.....	96
Table 27: Sobel filter vs. Robert filter	101
Table 28: Comparison of recognition results with different features	101
Table 29: Recognition results using CENPARMI and IFN/ENIT database.....	102
Table 30: Result of word spotting system after manual segmentation	107
Table 31: Comparisons of results on Word spotting	107
Table 32: Result of word spotting system on IFN/ENIT database	108

List of Abbreviations

BB	Bounding Box
CC	Connected Component
CCH	Center of Convex Hull
CENPARMI	Center for Pattern Recognition and Machine Intelligence
CH	Convex Hull
DTW	Dynamic Time Wrapping
ESL	End Shape Letter
GMM	Gaussian Mixture Model
HMM	Hidden Markov Models
HP	Horizontal Projection
HT	Hough Transform
MLP	Multilayer Perceptron
NLC	Non-Left-Connected
OBB	Overlapped Bounding Box
OCH	Overlapped Convex Hull
PAW	Part of Arabic Word

PCA	Principal Component Analysis
PR	Precision Rate
RBF	Radial Basis Function
RR	Recall Rate
SVM	Support Vector Machine
WST	Word Shape Token

Chapter 1

Introduction

Handwritten texts consist of artificial graphical marks and strokes that are written or carved by humans on a surface such as papers, metals, wood, glass, or rocks. The purpose of handwriting is to communicate, register, or transfer messages, news, ideas, information, and contracts. Handwriting is the basic tool that is used in many different areas and it is a skill that is learned by educators. This ability is affected by the educators' physical characteristics, age, personality, or mood. In addition, for the Arabic language, the skill might be affected by the educators' region, since various regions use different calligraphy styles. This explains the variability that is found in handwritten texts. The design of general handwriting related systems still remains a big challenge and an open problem in the area of pattern recognition and artificial intelligence.

Handwritten word segmentation, through the extraction of word units from the text and by finding word boundaries, is an essential task for many systems such as recognition and spotting. In this thesis, we look into the design and implementation of a system for word segmentation of unconstrained handwritten texts with no limitations in their writings such as texts that are not written in separate boxes, nor written with special pens, nor written neatly [129]. There are numerous challenges in the problem of word segmentation. We discuss these challenges and we present new solutions for them. Based on these solutions, new subsystems are developed and tested separately.

In this chapter, we define some essential concepts of handwritten document analysis in Section 1.1. The problem statement is given in Section 1.2. We discuss the motivation of our work in Section 1.3 while some of the Arabic characteristics are explained in Section 1.4. A brief comparison between Arabic and Latin languages is explained in Section 1.5. In Section 1.6, the challenges of Arabic handwritten texts segmentation are given and this thesis' objective is summarized in Section 1.7. The proposed approach is presented in Section 1.8 with the rationale behind its application along with our overall methodology. In Section 1.9, the contributions of this thesis are given. Section 1.10 describes the database used for this research while the outline of this thesis is given in Section 1.11.

1.1 Definitions

The recognition of handwritten texts is divided into offline and online systems. Online recognition refers to the techniques that deal with the automatic processing of handwritten texts using digitized instruments. In online recognition systems, the temporal information is available. Meanwhile, offline systems deal with the text's image.

There are some terms in handwritten documents analysis that may overlap with each other. Thus, these terms will be explained in detail. When dealing with handwritten word processing, these four concepts need to be defined precisely: word segmentation, word recognition, word spotting and word extraction or word separation. Word segmentation has two meanings in the literature. It may be used to refer to the process of dividing a word into either its characters or sub-characters. Moreover, word segmentation is used to refer to segmenting a text into words. Word recognition is the process of classifying the word from its overall shape. Word spotting, also referred to as indexing or searching, is a task to locate a word in a set of documents. Word extraction, known as word segmentation, aims at separating the text line into words. In fact, most of the authors use the term word segmentation instead of word extraction or word separation [94], [83], [73], [100], [144], [159],[95], [68], [121], [139], [133]. Therefore, we call the process of text line segmentation into words in this thesis, word segmentation.

In general, the algorithms dealing with word segmentation can be categorized into gap thresholding and metric classification. In the former, the segmentation is based on calculating the

distances between adjacent objects called Connected Components (CCs) in a text line and finding a threshold to distinguish between inter and intra-word gaps. In the latter, the gaps are classified into either inter or intra-word gaps by extracting some features and using classifiers.

1.2 Problem Statement

Words are the main building blocks in a text. In document understanding applications, the text needs to be segmented into word units. In the field of offline unconstrained handwritten document analysis, word segmentation is considered as a non-trivial problem to solve. Many difficulties arise in handwritten documents, making the segmentation process a challenging task, since word segmentation does not have much information about the text.

There are two main systems that are affected by the word segmentation's accuracy, namely recognition and spotting. In addition, the performance of these systems has direct effect on some applications such as document classification, translation, and scoring. In other words, if the words are wrongly segmented, all the systems' performance that are based on word level will be affected. In text recognition, there are two main approaches that address the segmentation problem called implicit and explicit. Segmentation and recognition are done simultaneously in implicit approaches, while the segmentation task is done before recognition in explicit approaches. These two methods, which are also called holistic and character-based approaches, are applied after segmenting the documents into words such as the work introduced in [67]. Thus, the output of these methods can be thought of as bounding boxes corresponding to each word in the text line [68].

For word spotting systems, many methods have been introduced for Latin languages and they reached promising results after segmenting the documents into words [102]. Two main approaches for word spotting, called template matching and shape code mapping, require word segmentation before spotting. For Arabic handwritten word spotting, only one work was introduced that applied spotting method after segmenting the documents into words [142]. This system got a low accuracy since the correct segmentation was low as well. The overall performance of correct word segmentation was 60% over only ten writers writing ten documents each.

1.3 Motivation

Arabic is the official language in more than 20 countries. In addition, it is the mother tongue of more than 300 million people, and one of the six formal languages in the United Nations [12]. Around the world, more than 1 billion Muslims read Arabic because it is their Holy book's language. The Arabic script was first documented in 512 AD. More than thirty languages use Arabic alphabets; some of them are Farsi, Pashtu, Urdu, and Malawi.

Handwriting still persists as a mean of information recording and communication in everyday life even with current technologies. A huge number of both modern and historical handwritten documents have been digitized to analyze, distribute and preserve them. Modern handwritten texts include bank cheques, postal addresses, forms, and contracts.

Extracting all the word images from a handwritten document is an essential pre-processing step for two reasons [73]. Firstly, for text recognition methods, which can be categorized into letter-based and word-based, there is a need to work on pre-extracted word images. Secondly, for word-spotting or content-based image retrieval techniques, all the word images in the documents are required to be pre-segmented properly. Most of the techniques in handwritten document retrieval and recognition fail if the texts are wrongly segmented into words.

Word segmentation is not only an important pre-processing step for word recognition and spotting but also for many other methods of Natural Language Processing (NLP). NLP is concerned with the interaction between humans and computers. Many areas of NLP require the word segmentation from handwritten documents to facilitate some tasks. For example, words need to be extracted to improve text-to-speech methods. Automatic summarization, translation, natural language understanding, part-of-speech tagging, text-proofing, text simplification, and automated essay scoring are researched tasks that deal with extracted words.

Large databases play an important role for the development of handwriting recognition systems. For evaluation, comparison and improvement of such systems, the text labeling (corresponding transaction) are expensive and time consuming. Word segmentation can improve ground truthing by transcription at the level of individual words.

Few methods have been proposed for Arabic texts segmentation in comparison to Latin-based languages. Arabic word recognition has received considerable attention in the literature. Recently, the exploration of Arabic word spotting in handwritten documents has begun [11]. However, only

five papers [22], [23], [142], [78], [53] have been published for word extraction from Arabic handwritten documents since separating texts into words is challenging due to the enormous different Arabic handwriting styles.

1.4 Arabic Characteristics

It is commonly accepted that segmentation and recognition of Arabic handwritten texts face some problems. Most of these difficulties are inherent to the nature of Arabic writing that are discussed in this section. Arabic characteristics are:

- Arabic script is written horizontally from right to left.
- Arabic script is either cursive or semi-cursive.
- Arabic alphabet contains 28 basic characters. Each character can have up to four distinctive shapes within a word depending on its position (beginning, middle, last, and isolated). Figure 1 illustrates the characters and their shapes.
- In addition to the Arabic alphabet, there are four non-basic characters, which are Hamza, Ta-marbota, Alif-maqsoura, and Madaa. In some papers, non-basic characters are classified as diacritics. There are different positions for the Hamza character. Hamza can be above or below character Alif, on characters Waaw or Alif-maqsoura, or isolated. Figure 2 illustrates the different positions of Hamza. Madaa can be situated above character Alif. Ta-marbota and Alif-maqsoura come at the end of a word either connected or isolated.
- Six characters cannot be connected from the left. They are Waaw, Alif, Daal, Thaal, Raa, Zaay, which we call non-left-connected (NLC) letters in this thesis. Figure 3 shows a word that has three different NLC letters.
- Each word may be composed of one or more Parts of Arabic Words (PAWs). In [126], a sub-word is defined "as being a connected entity of one or several characters belonging to the word". Figure 4 shows some words with different numbers of PAWs.

Name	Isolated	Beginning	Middle	End
Alif	ا	ا	ا	ا
Baa	ب	ب	ب	ب
Ta	ت	ت	ت	ت
Tha	ث	ث	ث	ث
Jeem	ج	ج	ج	ج
Haa	ح	ح	ح	ح
Khaa	خ	خ	خ	خ
Daal	د	د	د	د
Thaal	ذ	ذ	ذ	ذ
Raa	ر	ر	ر	ر
Zain	ز	ز	ز	ز
Seen	س	س	س	س
Sheen	ش	ش	ش	ش
Saad	ص	ص	ص	ص
Daad	ض	ض	ض	ض
Taa	ط	ط	ط	ط
Daad	ظ	ظ	ظ	ظ
Ayn	ع	ع	ع	ع
Gayn	غ	غ	غ	غ
Faa	ف	ف	ف	ف
Qaaf	ق	ق	ق	ق
Kaaf	ك	ك	ك	ك
Laam	ل	ل	ل	ل
Meem	م	م	م	م
Noun	ن	ن	ن	ن
Haa	ه	ه	ه	ه
Waaw	و	و	و	و
Yaa	ي	ي	ي	ي

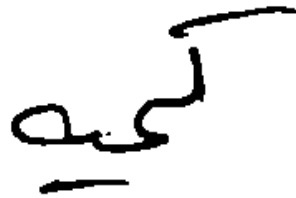
Figure 1: Arabic character shapes in different positions



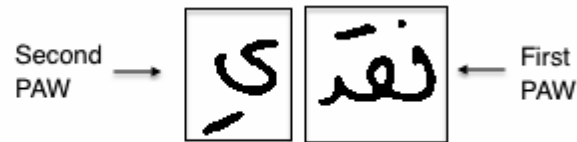
Figure 2: Hamza's positions



Figure 3: An Arabic word with three different NLC letters



(a) One PAW



(b) Two PAWs



(b) Three PAWs

Figure 4: Arabic words with different numbers of PAWs

- A PAW is composed of two parts, the main body and the secondary one which can be diacritics, non-basic characters or directional markings. In Figure 5, an illustration of the main and secondary bodies of a word is presented.

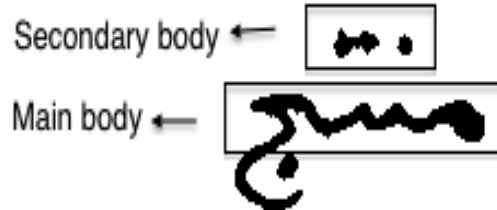


Figure 5: Main and secondary components of an Arabic word

- Diacritics are usually (composed of) one dot, two dots or three dots. Sometimes, two dots are written as a dash and three dots like ^ (logical conjunction symbol). Dots can help to distinguish the main bodies. In other words, one, two or three dots can differentiate two similar main bodies. For example, Daal, and Thaal (Figure 1) have the same main body and just one dot makes them have different sounds (constant). Ten characters have one dot, three characters have two dots and two characters have three dots. Dots may be placed above or under the letter's main body. Several representations of dots are presented in Figure 6.

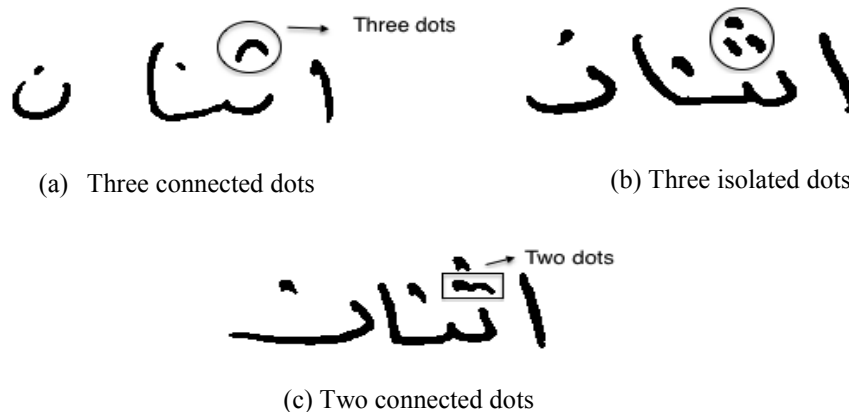


Figure 6: Some representation of dots

- One of the most important characteristics in Arabic writing is a baseline that is a horizontal line used to simplify and organize writing. Character connection usually occurs on this line. Figure 7 depicts the location of a baseline of the Arabic word.



Figure 7: A baseline of an Arabic word

- A PAWs' characters are normally connected on a baseline, but others can be connected vertically, which is common with some combination of characters such as Laam and Alif.
- Directional markings can be written above or below a character. In Figure 8, there is an illustration of all the directional markings. These directional markings may change the pronunciation and sometimes the meaning of a word. Figure 9 illustrates two words with the same letters and different directional markings, (a) means played, and (b) means toys. Directional markings cannot be combined within one character except with the directional marking Shadda.

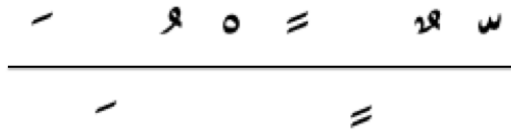


Figure 8: Arabic directional markings

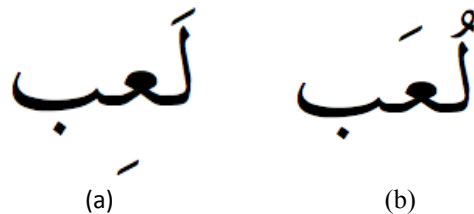


Figure 9: Same Arabic word with different directional markings

- Cursive Arabic writing has many styles, more than a dozen. The three main calligraphic styles are Kufi, Naskh, and Ruqaa. Some people use different calligraphic styles in their writing and sometimes within one word. Figure 10 shows some Arabic calligraphic styles.



Figure 10: Some Arabic calligraphic styles

1.5 Latin vs. Arabic

Generally, in a handwriting recognition process, the word is segmented into characters and then the classifier recognizes each character. However, character segmentation is not simple, especially in Arabic systems which have to confront many obstacles. The most important distinction among offline handwriting recognition methods in different languages is segmentation. It has been noted that a large number of recognition mistakes in handwriting recognition system are due to segmentation errors [43]. Most of the work in Latin script focused on character segmentation, which is considered easy in comparison to other languages. As stated in [30], it is commonly accepted that the letter segmentation for Latin cursive writing is still a problem that leads to the conclusion that letter segmentation in Arabic needs more research. Hence, researchers avoid letter segmentation while applying segmentation-free methods by recognizing the Arabic word as a whole. Thus, the words must be extracted before the recognition stage. Table 1 compares some aspects between Arabic and English languages to show the difficulties that might arise from such characteristics.

1.6 Challenges

There are many challenges in handwritten document segmentation. We can categorize the challenges into general problems, and Arabic-related problems. Both Latin and Arabic languages face many problems due to the common challenges of handwritten documents. Arabic-related problems are caused by some of its distinct characteristics.

The tasks of offline systems are considered harder than online ones, where the sequences of points and writing traces are measured. Offline systems are less accurate since only an image of a script is available. Offline systems can be divided into three categories: printed, historical and handwritten. Printed-related methods have achieved great accuracy, while most of the historical documents also get good performance. The difficulties involved in historical-related systems are mainly based on the pre-processing stage, not on segmentation since historical documents are usually written neatly considering the importance of the information given in such documents.

Table 1: Arabic vs. English

Characteristics	English	Arabic	Arabic Example
Size of character	Similar	No, because of ligature	بند - بند
Dotted characters	Only two	15 out of 28	
Number of dots change same body part	No	Yes	ت - ث
Position of dots change same body part	No	Yes, Above or below a baseline	تا - يا
Shape of letters based on location	Capital,(only names, and beginning of sentences) Small	beginning, middle, last, and isolated	عنصر أربعة مجمع ع
Non-basic characters	No	Yes	ء ~ ي ة
Variation of shapes for the same letter (with the same location)	No	Yes, due to the used calligraphic styles	انتها انتى انتها
Different writing styles of dots	No	Yes	⊙ - ⊙ - ⊙

However, the systems that deal with handwritten documents like recognition system and word spotting are more challenging because of the writing styles' variation and there is a need to perform many pre-processing tasks to improve the accuracy of such systems, one of these pre-processing tasks is segmenting the texts into words. Figure 11 shows the two types of recognition systems: online and offline along with three types of offline systems: printed, historical and handwritten. In offline handwriting systems, the main challenge is the individuals' writing styles. Generally, handwritten texts lack the uniform spacing that is normally found in machine-printed texts.

In the Arabic script, one of the major characteristics that differentiates this language from Latin-based ones is that twenty-two letters in the Arabic language must be connected on a baseline within a word. The remaining six letters cannot be connected from the left, which we call NLC letters. In this way, NLC letters separate a word into several parts depending on how many of these letters are included in a word. In other words, NLC letters indicate a separation of PAW. A study shows that NLC letters represent 33% of the texts [117].

Arabic texts have two types of spacing, intra-word gaps (gaps between PAWs within a word) and inter-word gaps (gaps between words). Intra-word gaps in the Arabic language are different from the ones in Latin-based languages. In Latin, intra-word gaps refer to the spaces that arise arbitrarily between any successive letters as a result of handwriting styles. In Arabic, in addition to the arbitrary spaces between letters as a result of broken PAWs, intra-word gaps are the ones between two PAWs, where the word must be disconnected due to NLC letters. This is part of the language's structure. Figure 12 shows intra-word gaps in both English and Arabic words.

In Arabic machine-printed texts, the inter-word gaps are much larger than intra-word gaps as illustrated in Figure 13. However, in Arabic handwritten documents, the spacing between the two types is mostly the same [26]. This is pointed out in Figure 14 from the CENPARMI cheque database [14]. Since the shapes of most of the NLC letters are curved, with the open end to the left, they are usually written with long strokes, which shrink the distance between words. Sometimes, they caused overlapping, or touching between words.

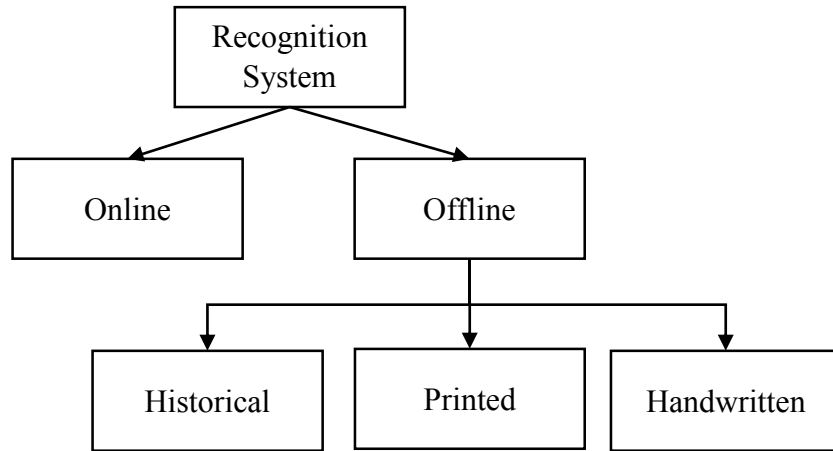


Figure 11: Types of recognition systems

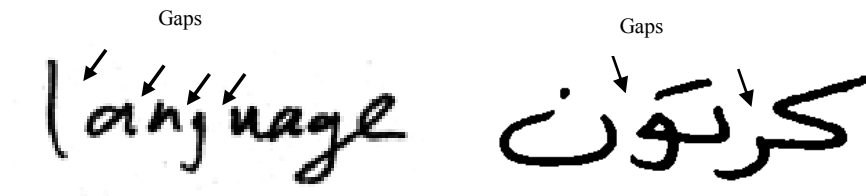


Figure 12: Intra word gaps

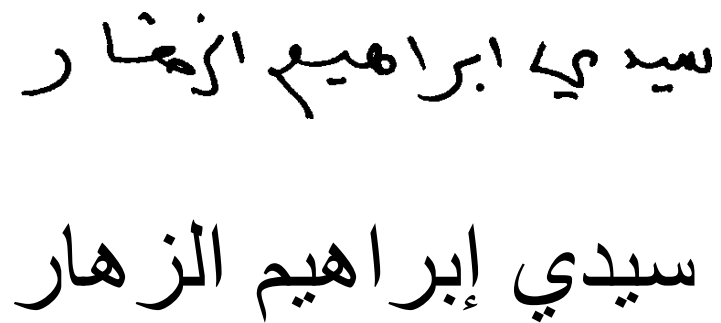


Figure 13: Printed (below) and handwritten (above) Arabic texts

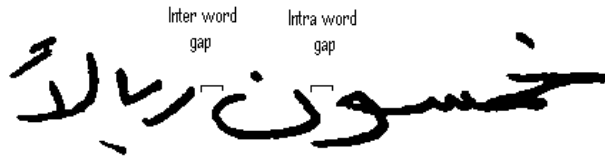


Figure 14: Intra and inter-word gaps in Arabic texts

1.7 Objectives

The existing methods are not adequate for extracting Arabic words from handwritten texts. In word recognition and spotting applications, it is very important to achieve high levels of accuracy. This research deals with the pre-processing steps and the words extraction of handwritten texts. Our main objective is to design an efficient and robust segmentation system to solve real-life and industrial problems. This thesis' research goals are as follows:

- A survey of offline handwritten word extraction.
- A review of the difficulties involved in word extraction.
- Propose a novel scheme for text segmentation based on end shape analysis and recognition.
- A survey of offline isolated handwritten letters recognition.
- Design a promising supervised learning system for isolated Arabic handwritten letters to enhance the segmentation process.
- A survey of baseline estimation methods for Arabic handwritten texts.
- A review of the difficulties involved in baseline estimation.
- Propose a robust baseline estimation method based on learning and feature extraction.
- Introduce an efficient approach to remove secondary components.
- Discover a promising supervised learning system for Arabic handwritten words to study the impact of word segmentation on word spotting system.

In general, we introduce new algorithms and techniques that can improve the accuracy of segmentation of Arabic handwritten words. We used all sources of foreground information and the knowledge of the language to improve the accuracy of segmentation.

1.8 Proposed Method

The main difference between our segmentation approach and previous methods is utilizing the knowledge of Arabic writing by shape analysis. In [22], [53], and [23], the authors pointed out the importance of using the language specific knowledge for Arabic text segmentation. Meanwhile in [103], the authors claim that one of the problems of Arabic text segmentation is the inconsistent spacing between words and PAWs. Our approach for segmentation is a two-stage strategy: (1) metric-based segmentation, and (2) recognition-based segmentation.

1.8.1 Utilizing the Knowledge of Arabic Writing

In the Arabic alphabet, twenty-two letters out of twenty-eight have different shapes when they are written at the end of a word as opposed to the beginning or in the middle. Two non-basic characters have different shapes at the end of a word. Therefore, analyzing these shapes can help identify a word's ending. In fact, there are just fourteen main shapes that can be used to distinguish the end of a word, since the remaining characters have the same main part but have a different number and/or dots' positions. Only NLC letter shapes are written the same way at the beginning, the middle or the end of a word. Therefore NLC letters cannot identify the end of a word. Consequently, End-Shape Letters (ESLs) can be categorized into two classes: endWord and non-endWord. Figure 15 shows the shape of the letter Noon when it is written at the beginning of the word, the middle and the end, and this letter is part of endWord class.

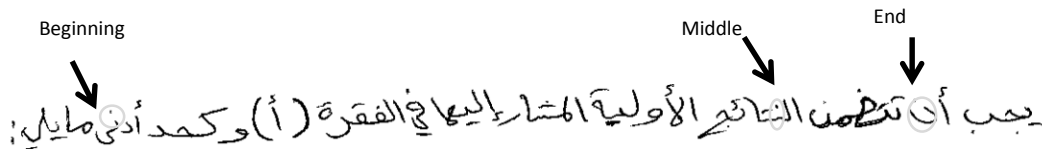


Figure 15: Letter Noon in different positions

1.8.2 Our Overall Methodology

Our methodology is composed of four main tasks: (1) secondary components removal, (2) baseline estimation, (3) metric-based segmentation, and (4) ESL-based segmentation. The input of our system is a text line image. The first subsystem aims at preprocessing the image and removing the secondary component that is explained in Chapter 2. The output of this subsystem are the main components of the text line. The baseline estimation task is described in Chapter 3. This method is a learning based approach that aims at determining the position of the baseline of the text line. The third subsystem is the first stage of the word segmentation technique which is described in Chapter 4. The purpose of this task is to pre-estimate the segmentation points between the words based on calculating the distances between the main components. Then the second stage of the word segmentation is explained in Chapter 7. The overall methodology is given as a block diagram in Figure 16. The details of our methodology are given in Figure 17.

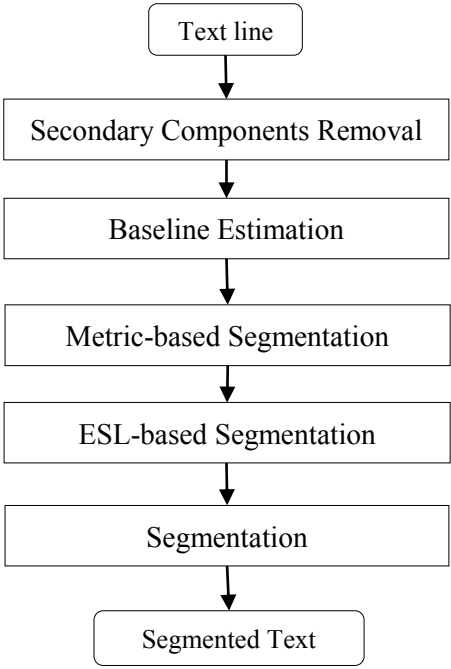


Figure 16: Overview of our methodology

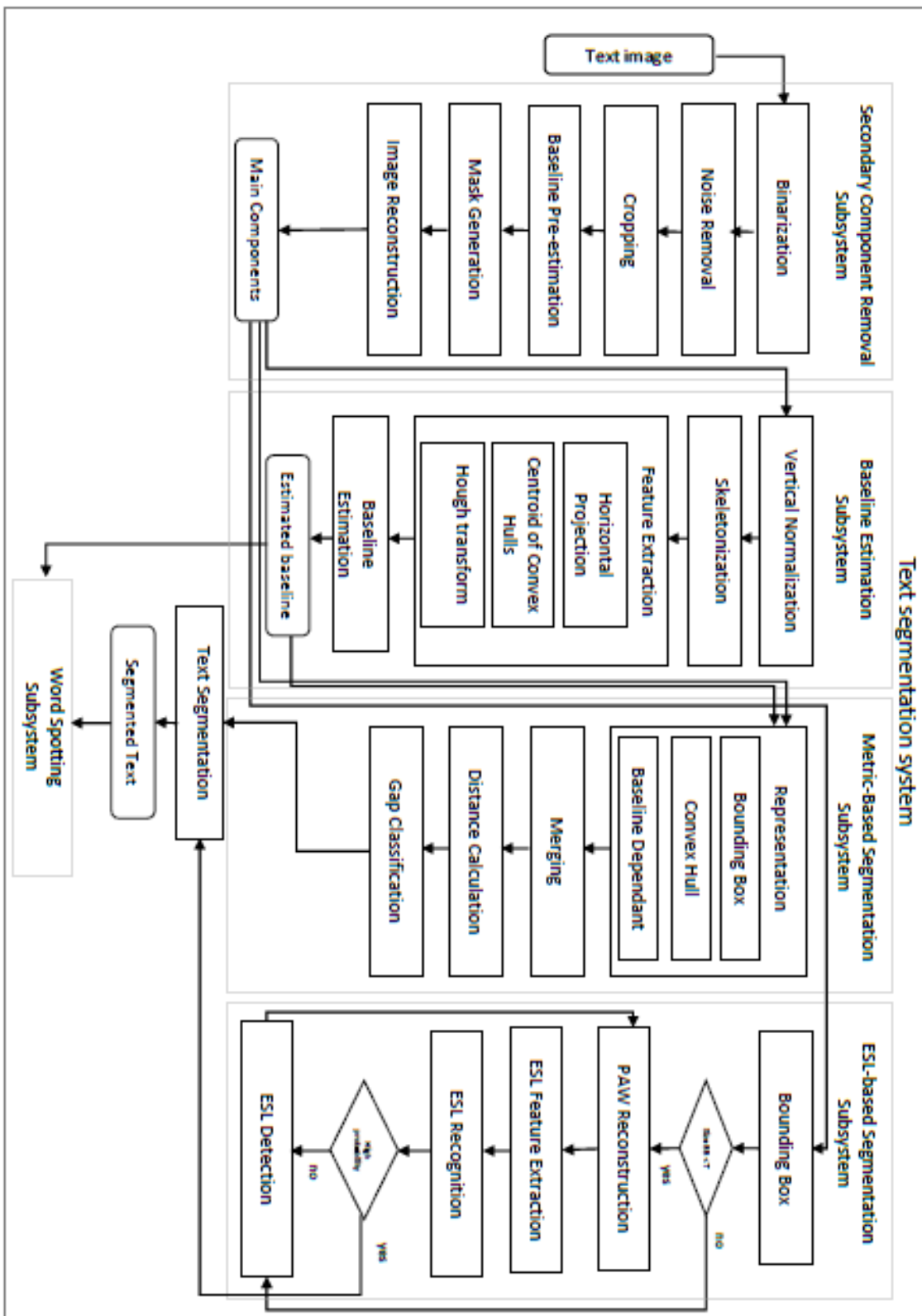


Figure 17: Overall methodology

1.9 Contributions

In this thesis, we present a coherent offline Arabic word segmentation system for multi-writer unconstrained scripts. The proposed system aims to solve the problem of lack of boundaries between words. The main contributions of this thesis can be summarized as follows:

- The introduction of a new word segmentation approach based on recognizing the last character of PAWs with advanced state-of-the-art technologies.
- The introduction of a novel learning-based baseline estimation method.
- The introduction of morphological reconstruction to remove secondary components to enhance the above processes.
- The creation of a new off-line CENPARMI ESL database.

1.10 Database

The Institute of communications Technology in Germany (IFN) and the École Nationale d'Ingénieurs de Tunis in Tunisia (ENIT) have developed an Arabic handwriting database. It contains more than 2273 handwritten forms from 411 writers with 26459 handwritten words. Most of the participants were familiar with the vocabulary. Each writer was asked to complete five forms, where each form contained 60 names. These forms were composed of 946 Tunisian town/village names. Some names appear more than 300 times while others were written only 3 times. Each handwritten word image comes with ground truth information that includes postal code, Arabic word in ISO 8859-6 code set, Arabic word as character sequence with shape index, number of words, characters and PAWs, baseline, baseline quality, and writer identifier, age, profession and writing quality. The database consists of five sets (a, b, c, d, e). Each set contains of about 6700 images. Table 2 shows the statistics of images with two and three words.

Table 2: Statistics of number of images with two and three words in IFN/ENIT database

Set	Two Words	Three Words
Set-a	842	185
Set-b	880	192
Set-c	211	21
Set-d	213	34
Set-e	178	8

1.11 Thesis Outlines

This thesis is organized into ten chapters, as described below:

- In Chapter 2, we discuss the work that has been done on secondary components removal. We propose our new approach of secondary component removal using morphological reconstruction. Experiments on the proposed method are also presented. The results are compared with the existing methods on the same database.
- In Chapter 3, we describe the previous works on baseline estimation for Arabic texts. We propose our own learning-based baseline estimation procedure. Different experiments are conducted and the results are presented.
- In Chapter 4, we review the studies on word segmentation for Latin-based languages. In addition, our metric-based segmentation algorithm is explained. The experiments and their results are described.
- In Chapter 5, a literature review is given for Arabic letter recognition methods. We present a complete isolated letter recognition system, discussing the different extracted features.
- In Chapter 6, we explore the created database of Arabic words with all the different shapes of the letters.
- In Chapter 7, we propose our new approach of word segmentation called End-Shape Letter Recognition based segmentation. Several experiments have been conducted and the results are compared with the metric-based segmentation method.

- In Chapter 8, our word recognition system is discussed and the results of extracting different features are presented.
- In Chapter 9, we define word spotting systems. Moreover, the impact of word segmentation on word spotting is discussed with the results of word spotting system.
- Finally, we summarize this thesis in Chapter 10 with some observations and directions for future works.

Chapter 2

Removal of Secondary Components

Using Morphological Reconstruction

In this chapter, we start by describing secondary components in Arabic language and the importance of their removal. In Section 2.2, the previous methods of secondary components removal are explained. Our proposed method, secondary components removal using morphological reconstruction, is described in Section 2.3. The experiments and their results are provided in Section 2.4. Finally, we conclude this chapter in Section 2.5.

2.1 Introduction

An Arabic word is composed of two parts: (1) main components that represent the primary part of the connected or isolated letters, and (2) secondary components (diacritics, dots, strokes, directional markings). In this thesis, the secondary components are removed to improve the performance of metric-based segmentation and baseline estimation. For metric-based segmentation, removing the diacritics can speed up the process by avoiding calculating the

coordinates of these strokes that do not have direct influence on the segmentation result(s). In fact, one of the problems of text segmentation is the existence of secondary components that overlapped in some cases with adjacent words. Several methods of baseline estimation are affected by these components such as horizontal projection, principal components analysis, contour following, and skeleton based methods. Though, the removal of secondary components avoids both the disturbance of the histograms in case of horizontal projection and principal component analysis and the error of points' selection of contour and skeleton based methods. However, many algorithms remove the secondary components to facilitate skew correction. Some methods also detect the secondary components to extract more features for recognition or spotting systems.

2.2 Related Work

Several methods have been applied that are based on height, area, positions of the components, binarization and thresholding, number of black pixels of each segment, bounding boxing, vertical layering, and contour following. The challenge is to apply a method based on the size of the connected components where isolated letters and secondary components are written in the same size, as seen in Figure 18. Generally, all of these methods are mainly based on the segments' size. None of these algorithms adopt the idea of restoring the words based on a roughly estimated baseline.

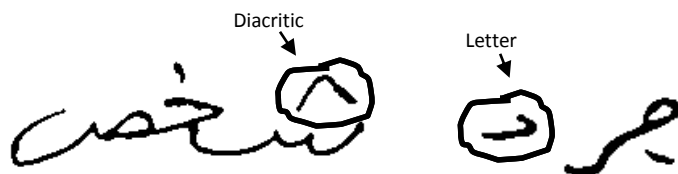


Figure 18: Main and secondary components with almost similar dimensions

In [38], the algorithm employs the following contour technique. The biggest segment is considered as the main component. Two steps are performed in [105] to remove diacritics. The first step filters the components relying on three criteria: size of the bounding boxes, area, and vertical layering while the second filter removes diacritics after estimating the baseline. In [32], they modified the algorithms proposed in [105]. This algorithm is based on the height, the area and the overlapping of the connected components. Chan et al. [41] remove the diacritics by binarizing the images. Then with some size and orientation ranges, a threshold was applied to the connected components. In [18], the technique is based on counting the number of black pixels of each segment and the number of rows included in the segment. The segment that has more than half the total number of the black pixels of the entire image is identified as a main component while the rest are considered as secondary. This method was applied to isolated characters of five different sizes and three fonts. In [104] and [54], the diacritics were removed based on the connected component's size. In [54], thresholds were determined based on empirical study.

2.3 Proposed Method

All the secondary components are written either below or above the main components, which are usually written on the baseline. The baseline occurs below the center of the image. We used this fact to extract only the connected components that were located in this position. Since the secondary components are concerned with the word image's middle area, we used a morphological reconstruction method by dilation that is based on an estimated baseband.

Our method is composed of two steps. At the first step, the secondary components are removed based on the components' sizes. At the second step, a pre-estimation of the baseline is calculated then the mask is generated based on the baseline. Finally, the reconstruction of the image is performed.

The reconstruction is a morphological transformation involving two images and structuring element that is used to define connectivity. We used 8-connectivity, which is a 3×3 matrix of ones with the center defined at coordinates (2,2). Morphological reconstruction processes one image, called the marker, based on the characteristics of another image, called the mask. The marker is the starting point for the transformation. In fact, the peak of high points in the marker image

identifies where the processing begins. The mask image constrains the transformation; hence the peaks spread out or dilate while being directed to fit within the mask image. The spreading processing continues until a stopping condition is reached. The fast hybrid reconstruction method is used [150].

Let I be the mask and F be the marker that are defined on the same discrete domain D and such that $F \subseteq I$. In terms of mapping, this means that;

$$\forall p \in D, F(p) = 1 \Rightarrow I(p) = 1$$

Let I_1, I_2, \dots, I_n be the connected components of I . The reconstruction of I from F denoted by $P_I(F)$ is the union of the connected components I which contains at least one pixel of F

$$P_I(F) = \bigcup_{F \cap I_k \neq \emptyset} I_k$$

We process only binary images. The word images are the masks. The marker is a generated binary image with the same size as the mask image with a horizontal line that is located below the middle of the image.

$$\text{Marker}(\text{Image}(\text{mean}(h):\text{mean}(h)+10, w)) = 1$$

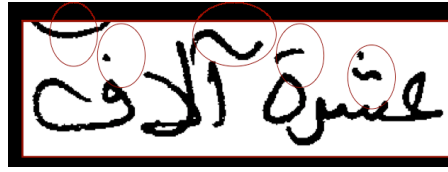
$$\text{Image} = \text{size}(\text{Mask})$$

where h and w are the image's height and width. The result of our method is illustrated in Figure 19.

We found another advantage in using our method: not only are the secondary components removed, but also some strokes were extracted near the edges. These strokes appeared because of the low performance of extracting the word image from bank cheques [14] or by the low performance of text line segmentation.

2.4 Experimental Result

We applied our method to the IFN/ENIT Arabic Tunisian city names database [127]. The training is done on randomly selected images from set-a. The experiments are conducted on 751 first images of set-a as in [32]. We used two metrics: false positives and false negatives. The false negatives are identified when the number of secondary components is less than the correct number;



(a) Binarization



(b) After removing secondary components based on size



(c) Below middle of the image



(d) Mask Generation



(e) Result

Figure 19: Result of our secondary component removal method

whereas the false positives are detected when a connected component is misclassified as a secondary component instead of a base form (PAW). Since the ground-truth information of the secondary components is not included with the IFN/ENIT, a manual evaluation is performed. The results were 2.90% false positives and 1.75% false negatives. The algorithm failed when the diacritics were touching the main components or the main components were written well above

the baseline as illustrated in Figure 20. Table 3 compares our method with the methods from Menasri et al [105] and Boukerma et al [32] on IFN/ENIT database.

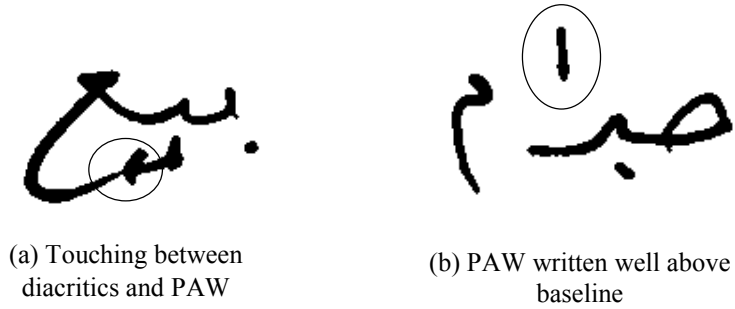


Figure 20: Error analysis

Table 3: Comparison of secondary components removal methods

Method	False Positives Detection	False Negatives Detection
Menasri [105]	14.24%	5.32%
Boukerma [32]	5.72%	7.05%
Proposed Method [76]	2.90%	1.75%

2.5 Conclusions

We believe that the preprocessing stage can improve both the recognition and the spotting systems. In this chapter, a preprocessing method of removing secondary components for Arabic handwriting texts is presented. Our proposed method is based on the use of morphological reconstruction. After binarizing the image, using state of the art technique, some secondary components were removed

based on a threshold. Then, an initial estimation of the baseline is calculated to facilitate the mask generation. Finally, the image is reconstructed based on the generated mask. The method is evaluated by using a standard database and is compared with two previous algorithms.

Chapter 3

Learning-based Baseline Estimation

Baseline estimation is an important pre-processing step in Arabic text recognition systems. In this chapter, the text baseline's definition is presented in Section 3.1. The motivation and the concerns about this essential preprocessing task are given in Section 3.2. Next, the baseline error measurement is included in Section 3.3. The challenges that are related to Arabic writing characteristics are described in Section 3.4. The previous work on baseline estimation for both Latin and Arabic languages are provided in Section 3.5. In Section 3.6, the database used, the generated database for the proposed method and our method are described. Finally, our experiments are also presented in detail in Section 3.7. This chapter is concluded in Section 3.8.

3.1 Baseline Definition

Arab people use an imaginary line called the baseline, which is the main property in an Arabic script, in an un-ruled paper to simplify the actual writing. The baseline concept does not have a precise definition, but it can be defined as "the virtual line on which cursive or semi-cursive writing characters are aligned and/or joined" [17]. In other words, it is the line at the height where letters are connected, so the main bodies are written above it except for the descenders. In fact, the Arabic

text can be split vertically into three regions: upper, lower and middle. The main part of the letters, loops and their connections are located in the middle region which is part of the baseline position. Meanwhile, ascender, descender, dots, and diacritics lie either in the upper or/and lower parts. Figure 21 shows the baseline's main properties.

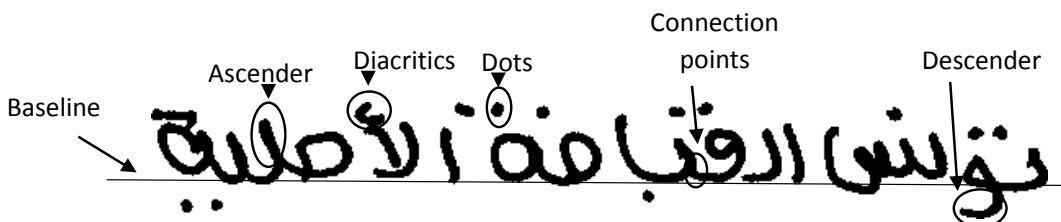


Figure 21: Main properties of baseline

3.2 Motivation

In Arabic printed texts, the baseline can be detected easily by finding the row that has the most number of black pixels. However, in handwritten texts, this procedure cannot be applied due to the extreme variation of writing styles and irregularity in PAW alignment. Baseline estimation is used for skew normalization, slope correction, for segmentation [107], and for feature extraction [125], [51]. The dots and their positions, which are below or above the baseline, along with word descenders and ascenders, can be identified by their baseline positions. Moreover, baseline provides important information regarding text orientation as well as the connection points between characters. Baseline identification has direct influence on recognition accuracy [117] and segmentation performance. The failure of such systems may be caused by inaccurate estimation of the baseline. Moreover, some approaches are used in baseline detection as a main key for text line separation [120], [29], and [119]. In this thesis, the baseline estimation method is used for metric-based segmentation and for feature extraction stage of word recognition.

3.3 Baseline Error Measurement

The baseline error measurement is explained in [126]. To rate the baseline position, they prepared a survey with hundreds of Arabic handwritten words from IFN/ENIT database with marked baseline positions. Then, a group of Arabic native speakers were asked to tag all baseline positions as “excellent”, “acceptable”, and “insufficient”. They observed that the baseline position was evaluated as excellent with up to a 5 pixel vertical position error, the baseline position was evaluated as acceptable for up to a 7 pixel vertical position error, and when the vertical position error is more than 7 pixels the baseline position was evaluated as insufficient. To evaluate our algorithm, we have calculated the distance between the estimated baseline with our algorithm and the baseline positions in the ground truth. This distance is used for performance evaluation.

3.4 Challenges

Some characteristics of Arabic writing are challenging for baseline detection. Diacritics, non-basic characters and directional markings can either affect the accuracy of baseline estimation, since baseline methods are concerned with the main body of the word, or the speed of the process that can be affected when removing such strokes. In addition, the variation of baseline within a word (between PAWs) has a big effect on baseline methods performance. A short description of some of the challenges can be found in [17], [116] and [115]. The challenges that are related to the language can be summarized as follows:

- Secondary components: strokes that have zigzag shapes or long diacritics. (Figure 22(a))
- Word slope. (Figure 22(b))
- Overlapping
 - Inter-overlapping means some characters from different words are overlapped. (Figure 22(c))
 - Intra-overlapping means some characters within a word are overlapped. (Figure 22(d))
- Text line length
 - Long text line with misaligned PAWs. (Figure 22(e))

- Short text line with small PAWs. (Figure 22(f))
- Ligature
 - Long ascenders or descenders. (Figure 22(g))
 - Many ascenders or descenders. (Figure 22(h))
 - Touching ascenders or descenders. (Figure 22(i))

Some of these challenges are combined, making the problem more challenging e.g. short word with slope. In addition, baselines vary among different writers. Most of the proposed methods reach satisfactory results for long lines text, but they are not as accurate with lines containing one or a few words. Several methods remove secondary components [37], [15] to improve the performance of baseline estimation. However, the rest of the challenges are not easy to manipulate before baseline estimation.

3.5 Related Works

Several methods have been proposed for baseline estimation. The various methods of Arabic baseline estimation in the literature can be categorized by (a) the basic entity of estimation (text line, word, or PAW), (b) the information of the baseline's representation (e.g. skeleton, contour), and (c) the restriction required by the technique.

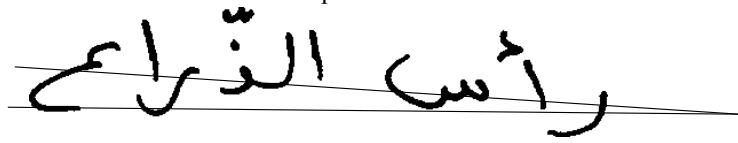
In general, baseline estimation techniques can be divided into three main categories: (1) statistical distribution methods, (2) geometrical analysis methods, and (3) combination of (1) and (2). The first technique is based on the foreground pixels distribution, while the second mainly relies on baseline-relevant points' selection. Arabic baseline detection is gaining more attention due to the reasons mentioned earlier about the importance of features that are related to the baseline for Arabic texts.

Long diacritics



(a)

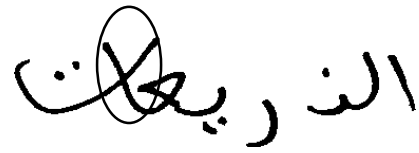
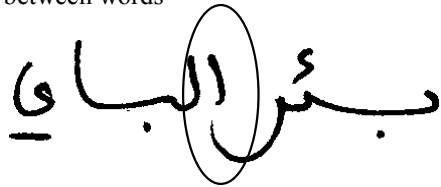
Slope



Overlapping
between words

(b)

Overlapping
within a word



(c)

(d)

Misaligned PAWs

Small PAWs



(e)

(f)

Long descender

Many descenders



(g)

(h)

Touching descenders



(i)

Figure 22: Baseline estimation challenges

In current, the state of the art methods is Horizontal Projection based approach. This method analyses the density histogram by counting the number of foreground pixels for each row, while assuming that the maximum number of elements on a horizontal line would include the baseline. The first attempt for Arabic baseline estimation by Parhami and Taraghi used the horizontal projection in 1981 [45]. In [113], Nagabhushan et al. proposed a piece-wise painting schema where black and white blocks are extracted from the text line. After removing dots and diacritics, a horizontal projection was calculated for black blocks. Based on maximum horizontal projection profile, candidate points were selected. Olivier C. et al. [117] assumed that all the words are perfectly horizontal or have a small inclination. They applied horizontal projection while taking into account the position of the loops. El-hajj et al. [51] detected the upper and lower baselines which are based on horizontal projections. The baseline is detected through iterations of changing angles and horizontal projections [15]. The highest peak in the projection, located below the middle line, is assigned as the baseline [55]. A horizontal histogram was combined with directional features based on a skeletonized PAW [5], [4], and [6]. Several steps have been implemented: binarization, connected components extraction, dots and diacritics removal, horizontal projection and pre-estimated baseline regions, feature extraction, and baseline detection.

In [97], the entropy method was applied to measure baseline relevant information. The histogram density and corresponding entropy were calculated for each projection. Petchwitz et al. proposed an enhanced horizontal projection method [126]. The binary word image was transformed into Hough space. The dark regions of this space indicate line directions with black pixels on a straight line. The maximum in the Hough space identify the baseline position. Principal Component Analysis (PCA) is a statistical procedure to find the directions, called principal components, along which a distribution exhibits the greatest variation was used in [37]. The baseline estimation was determined according to foreground and background pixels. After angle detection and baseline estimation using eigenvector (by choosing the eigenvector with the largest eigenvalue), the image is rotated so that this estimated baseline angle lies horizontal. A horizontal projection was applied to find the peak as the baseline. The experiments were done with and without diacritics. Generally, horizontal projection profile is robust and easy to implement, however any statistical approach needs long straight line of text, which is not the case in handwritten documents, since the researcher's assumption is based on that the density of pixels is higher around the baseline position. Thus, horizontal projection histogram based methods fail in

estimating the baseline with short text line and text having great number of ascenders, descenders and large diacritics. Moreover, PAW misalignment caused some errors on baseline estimation. In addition, horizontal projection is very sensitive to skew.

Various researchers employ word skeletons and word contour processing for baseline estimation. Pechwitz et al. [125], and [126] extract many features from word skeleton and categorized these features into relevant and irrelevant baseline features. As a consequence, a pre-estimated baseline region was determined based on the irrelevant features. The relevant baseline features that are located in the pre-estimated baseline region are extracted. Finally, a regression analysis of the selected features was completed to estimate the final baseline position. Boukerma et al. [31] presented an algorithm based on skeletonized PAW processing. Some points are grouped based on the aligned trajectory neighborhood direction. Next, many topological conditions were applied on the set of points that were found in the first stage to estimate the baseline. In [57], the method was based on the contour's local minimum points. After locating the points where the contour changes direction from the lower to the upper of the image, two step linear regressions were applied to find the baseline position. Errors would occur with this kind of approach, when incorrect points were extracted due to large diacritics or word with small PAWs. Sometimes connection points between characters are not laying over the baseline. In fact, complex calculations were needed for this time-consuming operation.

In [32], an algorithm was proposed that combined the different representations of skeleton, and contour, with different techniques such as horizontal projection method, linear interpolation, and some heuristics. Ziaratban et al. introduced a baseline estimation method using a template matching algorithm with a polynomial fitting algorithm [156]. In [49], a Hough transform method was applied on main component of PAWs. A data-driven baseline detection was introduced in [140]. The main idea is to use data-driven methods trained on local character features to find the probable baseline region. Gaussian Mixture Models (GMMs) were used to estimate the likelihood of each baseline region. Finally the horizontal projection was applied on this region. Voronoi Diagram, defined as “lines that bisect the lines between a center point and its surrounding points”, was used in [16]. Edge detection, contour tracing and sampling were used to measure Voronoi diagram points. Then, only the horizontal edges were used with rule-based approach.

In Latin languages, the horizontal projection method is used to estimate lower, upper and baseline for words [33], and [36]. Most studies in the literature use some heuristics based on

horizontal projection profile [27], and [69]. Moreover, the horizontal projection that is based on contour pixels is proposed in [8], [29], and [152]. The baseline was estimated by maximizing the entropy of the horizontal projection histogram [151]. Linear regression is used in different approaches. A pseudo convex hull from mathematical morphology with weighted least squares was employed [59]. In [66], least square linear regression was applied by Graves et al. [66] and by Modolo et al. [34]. Two linear regressions through the minima and maxima were used by Dalal et al. [46], and Liwicki et al. [93]. Some researchers use contour based analysis combined with heuristics [20], [122], [39], and [40], and others classify the local extrema of contours by a multilayer perceptron (MLP) [155], [65], and [24]. A polygons analysis was investigated by Bengali [135]. An approach was introduced by Simard et al. [124] to automatically detect a set of points and classify them by machine learning techniques. In [70] and [71], an approximating cubic splines function was employed. A heuristics approach based on local Extrama was used in [39].

Since we use IFN/ENIT database for our experiments, Table 4 summarizes the reported results of baseline estimation based on this database. The entity refers to the basic unit that is used for estimating the baseline. Much information is added to this table, like the number of images used for testing, the name of the sets from IFN/ENIT, the word representation such as skeleton or contour, the techniques that were applied, the result (“<” refers to the number of pixels below the acceptable error range), and the type of error. We classify the errors that are related to baseline estimation into two types: (1) density confusion, and (2) miss determined candidate baseline–relevant points. By density confusion, we refer to the errors that occurred when the techniques are mainly based on both the distribution of the foreground pixels in the image, and the assumption that the high density is on baseline area. This assumption is true in case of long text line, but it does not apply to a word, a few words or short text line. The second type of errors arise when some points, considered as indication of the position of the baseline, were wrongly selected, and the technique used was based on these candidate points. Table 5 shows the comparisons between the previous works on baseline estimation and the challenges of Arabic writing characteristics that they are affected by.

Table 4: Results of some methods reported in the literature

Method	Entity	Images	Set	Representation	Technique	Result	Error
[31]	PAW	2740	a	Skeleton Contour	Linear interpolation	< 10 : 69.11	2
[156]	PAW	2700	a		Template matching	< 10 : 85.33	2
[37]	PAW	1000		Skeleton	PCA Horizontal Projection	< 7 : 82	1
[125]	Word	26459		Skeleton	Linear regression	< 15 : 95	1
[97]	Word	6567	a		Horizontal Projection	< 7 : 66.3	1
				Contour	Horizontal Projection Heuristics	< 7 : 74.3	1
				Contour	Heuristics Feature selection	< 7 : 82.3	2
				Contour	Entropy	< 7 : 52	2
				Skeleton	Heuristics	< 7 : 87.5	2
					Hough transform	< 7 : 71	2
[126]	Word		all		Horizontal projection Hough space	< 7 : 82.8	1
				Skeleton	Feature selection Linear regression	< 7 : 87.5	1

Table 5: Challenges related to baseline methods

Type	Statistical Distribution			Geometric Analysis				
Methods Challenge	Horizontal Projection	PCA	Linear regression	Hough transform	Template matching	Voronoi Diagram	Word Skeleton	Word Contour
Small dots	Yes	Yes	Yes	No	Yes	No	No	No
Large diacritics	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Slope	Yes	Yes	Yes	Yes	No	No	No	No
PAW alignment	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Short text	Yes	Yes	Yes	Yes	No	Yes	No	No
Long text	No	No	No	No	No	Yes	Yes	Yes

3.6 Proposed Method

We propose a supervised learning method for baseline estimation. We aim to use a learning technique in the preprocessing stage instead of relying on geometric heuristics that need expert knowledge for designing features while sometimes lacking robustness. On the other hand, learning based approaches allow automatic features to be learned from input images. This method is depicted in Figure 23. The subsystem is divided mainly into three parts. The first part is related to the preprocessing which is concerned with obtaining a compact and reliable representation. The second part is related to the feature extraction process while the third is related to the estimation process.

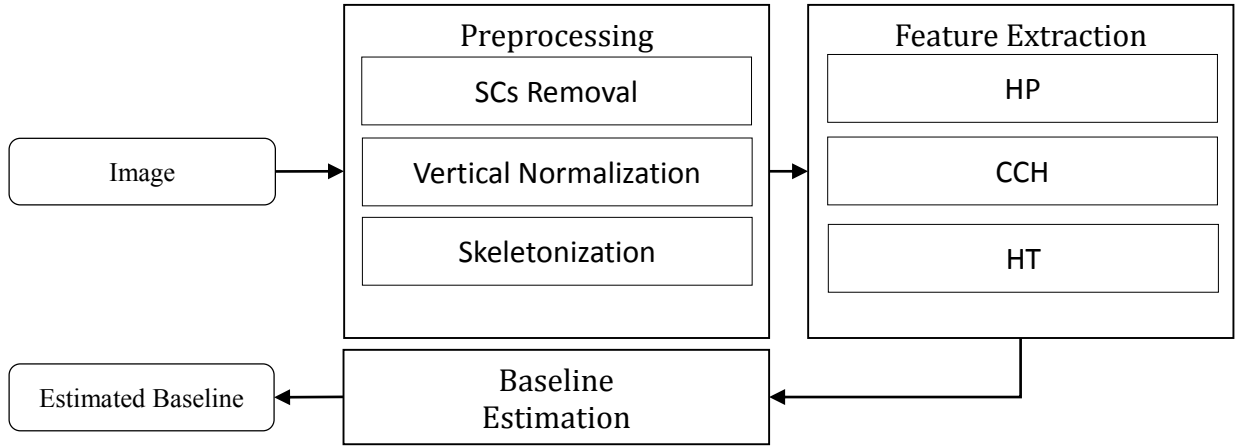


Figure 23: Proposed method

3.6.1 Our Method

The baseline of an Arabic word consistently lies below an image's middle line [59]. After analyzing some images from the IFN/ENIT database [127], we observed that the baseline of the words, after cropping and normalization, was roughly between the 30th and 90th row pixels along Y axis of the image. Figure 24 shows the baseline's range. As a consequence, the number of baseline classes is limited. For supervised learning of handwritten words' baseline positions, we need some handwritten words as training samples with their baseline positions. The IFN/ENIT database is used since they include the baseline positions in the ground truth. These baselines were used to train the classifier. Let $W = \{w_1, w_2 \dots, w_n\}$ be a collection of training word samples. $F = \{f_1, f_2 \dots, f_m\}$ contains baseline-dependent features of a word while $B = \{b_1, b_2 \dots, b_j\}$ are the baseline classes. Our training model consists of P , a set of baseline-dependent features that include the position of the baseline.

$P = \{(w_i, f_{i1}, f_{i2}, f_{i3} \dots, f_{im}, b_j) \mid w_i \text{ is a word, } f_{i1} \text{ to } f_{im} \text{ is a set of features for word } i, \text{ and } b_j \text{ is the position of a baseline}\} \quad (3)$

where $i = 1 \dots, n,$
 $f_{i1}, f_{i2}, f_{i3} \dots f_{im},$
 $0 \leq b_j \leq \text{number of baseline classes.}$

First, the secondary components are removed as discussed in Chapter 2. Then, since our interest is focused on the rows of an image, the images were only normalized vertically. The

normalized image sizes were 128 x M, where M is the original width of the image. After normalization, all the images are skeletonized.

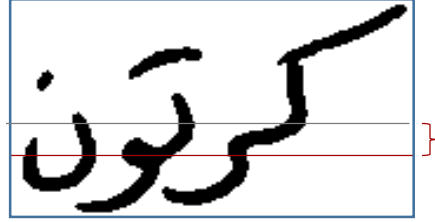


Figure 24: Baseline range

The performance of the classifier depends on the quality of the features. Some baseline dependent features have been extracted to facilitate baseline estimation. The three features that have been used are: (1) horizontal projection (HP), (2) centroid of CH, and (3) center of line segments using Hough Transform (HT). These features are discussed in detail below.

- The horizontal projection of an image L is given by:

$$HP(i) = \sum L(i,j)$$

where HP(i) is the HP of row i,
and L(i,j) is the pixel value at (i,j).

The number of the horizontal histogram features extracted is 128. The HP of an Arabic handwritten text image is shown in Figure 25.

- After extracting the main components, the CH is generated for each main component. Then, the center of each CH (CCH) is calculated. Finally, the mean of the centers is given below:

$$CCH = \frac{\sum C(i)}{n}$$

where C(i) is the center position of the CH of a main component,
and n is the number of main components.

Figure 26 illustrates the center of the CH of each connected component and the mean of the centroids.

- We extract horizontal line segments using HT method. The HT detect lines using the parametric representation of a line:

$$h = x * \cos(\theta) + y * \sin(\theta)$$

where h is the distance from the origin to the line along a vector perpendicular to the line, and theta is the angle between the x-axis and this vector.

A parameter space matrix is generated whose rows and columns correspond to h and theta values, respectively. After we compute the HT, we find peak values in the parameter space. These peaks are represented by horizontal lines in the input image. After identifying the peaks in the HT, we find the endpoints of the line segments corresponding to peaks in the HT. Figure 27 shows the extracted line segments using Hough transform. After extracting the line segments, we calculate the mean of these segments. We use this feature only for the second experiment.

The training and testing models are generated using Support Vector Machine (SVM) that is described in Section 5.4.

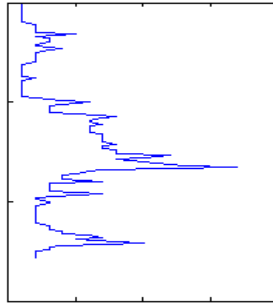


Figure 25: Horizontal projection of text image

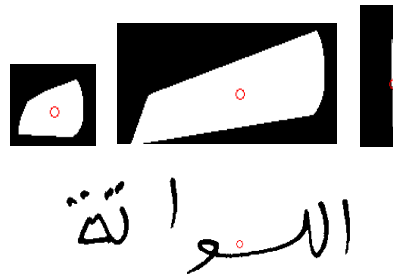


Figure 26: Centroids of convex hulls

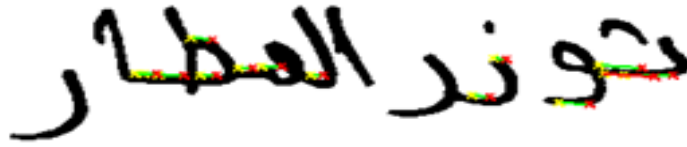


Figure 27: Line segments using Hough Transform

3.6.2 Baseline Ground Truth

The IFN/ENIT database [127] has more information in the ground truth that is not commonly found in other databases. It has two special features that are called baseline position and quality flag. A pre-baseline positions were estimated for each word image and they are documented in the truth file under BLN label (BLN: Y1, Y2) and generated automatically. Then, they are verified manually. The baseline position was adjusted during the verification procedure. While verifying whether the baseline could not be corrected because of handwriting styles' variation, the quality flag for the baseline was marked as “bad” instead of “acceptable”. In addition, the writing quality could be set to “bad”. This baseline ground truth is very essential because we are able to evaluate spontaneously our baseline estimation algorithm on the basis of a quite large database.

3.6.3 Baseline Database Generation

As mentioned earlier, the baseline positions are documented in the ground truth files. Thus, we use this information to generate the database for our baseline estimation method. We conduct two types of experiments.

The first experiment is based on a survey [42], the authors observed that for up to a 5 pixel vertical position error, the baseline was evaluated as excellent. Based on this observation, the difference between each two consecutive classes is 5 pixels. We divided this experiment into three parts based on the number of words per image. Thus, the number of classes varies between 11 and 14. The images were distributed in the classes based on the baseline position in the ground truth after normalization.

For the second experiment, we use the midline position to assign each word an image to a class. However, we use the range for evaluation. In other words, each word image is included in a class based on its midline baseline position.

3.7 Experimental Results

In the first experiment, which is described in [75], set-a is used for training and set-b for testing. Each set in the database is divided into three parts, based on the number of words per image (set-one, set-two, and set-three), in order to examine the performance of our baseline estimation approach based on the number of words per image. Figure 28 shows the distribution of the words in the images for set-a, and set-b. In this experiment, the evaluations are divided into three separate parts, for each sets, set-one, set-two, and set-three, since each set has different number of classes. We found that the fewer number of words, the fewer number of classes. Table 6 summarizes the results of our approaches after extracting two features: horizontal projection and centroid of CHs.

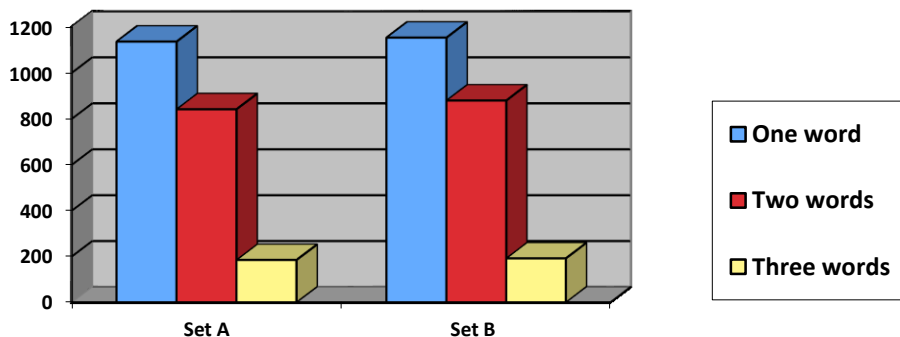


Figure 28: Distribution of the images with respect to the number of words

In the second experiment, we use set-a, set-b, and set-c for training, and set-d for testing. Since we use a midline of the baseline position (not in a range), the number of classes is extended. We use the midline position of the baseline instead of the 5 pixel range because some classes might be discarded during evaluation. Though, we need more data for training. The database consists of

74 classes but some classes have few images that affect the system's performance. The classes that are composed of few images are the ones have text with extremely high or low baseline position. The number of training samples is 8827 images, and the testing set contains 871 images. Table 7 shows the result of our method using 74 classes and two features: horizontal projection, and centroid of CHs. Some of the classes contains only 7 images.

Since most of 74 classes consist of few images, we apply some experiments on the classes that that contains lots of images. The total number of these classes is 21 classes. Our method's result in using 21 classes with different extracted features is given in Table 8.

The baseline ground truth in IFN/ENIT database allows intensive evaluation of baseline estimation methods. Our approach uses supervised learning technique that enables us to reach acceptable baseline (≤ 7) in 98.7% of the testing samples. Most of the errors are caused by the presence of descenders' long strokes as seen in Figure 29. Table 9 shows the comparisons between several methods that use IFN/ENIT database in their experiments.

Table 6: Baseline estimation results of the first experiment

Class	Error in pixels	Percentage
One word	≤ 5	47.21
	≤ 10	90.39
Two words	≤ 5	18.15
	≤ 10	30.70
Three words	≤ 5	15.22
	≤ 10	47.28

Table 7: Result of 74 classes

Error in pixels	Percentage
≤ 5	74.76
≤ 7	82.23
≤ 10	89.03
≤ 15	94.63
≤ 50	100

Table 8: Results of learning-based baseline estimation

Error in pixels	Percentage HP	Percentage HP + CCH	Percentage HP +CCH+ HT
≤ 5	96.10	96.27	96.83
≤ 7	98.56	98.7	96.85
≤ 10	99.69	99.77	96.86
≤ 15	99.98	100	96.86
≤ 50	100	100	96.86

Table 9: Comparison among baseline estimation methods

Method	Used Set/s	Used images	Baseline error in pixels				
			≤ 5	≤ 7	≤ 10	≤ 15	≤ 50
[31]	a	2,240	30.89	--	69.11	87.19	--
[156]	a	2,700	74.56	--	85.33	91.82	--
[37]	a	1,000	--	88	--	--	--
[97]	a	6,567	77.8	82.3	--	--	--
[126]	a, b, c, d	26,470	76.7	87.5	94.1	97.5	100
Our method	a, b, c, d	9,698	96.27	98.7	99.77	100	100

بکند اعة

Figure 29: Error of baseline estimation

3.8 Conclusions

After utilizing existing methods for vertical normalization and skeletonization, we applied a learning-based method using some baseline relevant features such as horizontal projection with a selection of feature points. Although we use a few features, our method shows promising results. The future work will include slant correction to examine the effect of such preprocessing in our method. In addition, more features can be extracted, like the minimum variance, locating holes, rotating the image to find the peak of horizontal histogram, and extracting the features from PAW.

Chapter 4

Metric-based Segmentation

In this Chapter, we present the first stage of our proposed methodology: the metric-based segmentation method. We apply this method for two reasons. For the first one, our proposed idea for segmenting the text that is based on recognizing the final characters cannot be applied on six characters, so we need additional procedures to avoid such cases/failures. The second reason is the ability to compare our approach to some of the state of the art methods on the same database. We discuss the main algorithms that are used to segment the text lines into words in Section 4.1. The related work of word segmentation for Latin-based language is presented in Section 4.2. The method that we use is explained in Section 4.3, followed by the experiments in Section 4.4. The conclusion of this chapter is given in Section 4.5.

4.1 Introduction

A wide variety of word segmentation methods for handwritten documents have been reported in the literature. Algorithms dealing with segmentation are mainly based on the analysis of geometric relationship of adjacent CCs. The common assumption for word segmentation methods is that a pair of adjacent components that are part of the same word are significantly close to each other. In

another words, inter-word gaps are much larger than intra-word gaps. In general, the work for the problem of word segmentation differs in two aspects [94]: (1) the way the distance is calculated between adjacent CCs and (2) the technique used to classify the calculated distances. The algorithms that are used can be categorized into metric-based and classifier-based. In the former, the threshold is determined to distinguish between gap types [102]. In the latter, the gaps are classified into either inter or intra-word gaps based on the extracted features [147].

4.2 Previous Work

In this section, we review the previous works completed for the two main methods: metric-based methods and classifier-based methods. Moreover, we discuss some methods that were applied on historical documents. Finally, an overview of the contests that have been organized for handwriting segmentation methods with their results is given in Section 4.2.4.

4.2.1 Metric-based Segmentation Approaches

Some distance metrics were defined in some related works. Threshold is determined based on either heuristics or learning techniques. Seni and Cohen [83] were the first to discuss the word segmentation problem. Eight distance metrics were presented. These metrics are the bounding box, the minimum and average run-length, the Euclidean and several combinations of them which depend on heuristics. Mahadevan et al. [98] introduced a new metric called convex hull. This metric was compared with the bounding box metric, both run-length and Euclidean metric. Convex hull based method showed better performance. Verga et al. [149] incorporated a tree structure technique for word extraction. The decision about a gap was taken in terms of both a threshold value and its context such as the relative sizes of the gaps' surroundings. Kim et al. [88] used the module that was described in [137]. The work of Marti and Bunke [103] were based on the convex hull metric and a threshold for gap classification. The threshold was defined for each text line. Louloudis et al. defined a procedure for segmentation that is divided into two steps [94]. In the first step, after a slant correction, the distance between adjacent CCs were measured using

Euclidean distance. The gaps were classified based on a global threshold in the second step. The global threshold was based on the black to white transition in the text line. In [121], they calculated what they called SVM-based gap metric to formulate the distance between adjacent CCs. A global threshold was used to classify the gap. This method was enhanced in [139]. A normal distribution for each class was formulated based on initial classification. Then, the candidate gaps that lie around the threshold were reclassified by employing the maximum likelihood criterion. Distances between CCs were computed based on the Delaunay graph [90]. A k-mean was used to determine the inter and intra-word gap. Louloudis et al. [94] used a combination of two metrics namely, convex hulls and Euclidean distance. They make use of the Gaussian mixture theory to model two classes. Kurniawan et al. extracted the contour of the words and the threshold is computed from median white run-length [102].

4.2.2 Classifier-based Approaches

Some methodologies use classifiers for the final decision, either between words or within a word. In [87], a simple neural network was adopted to determine the segment points. The neural network used eight input units, four hidden units and one output unit. Input parameters to the neural network were properties of bounding boxes, the center lines of bounding boxes, intervals between center lines and height of bounding boxes. Huang et al. [73] presented a work that uses a three-layer neural network after extracting eleven features. These features include seven local features such as height and width of CCs along with four global features like the average height of CCs. The neural network had eleven input units, four hidden units and two output units. Luthy et al. [96] considered the problem of segmentation as a recognition task. At each position of a text line, it was decided whether the position belonged to a letter or to a space between words. Three recognizers based on Hidden Markov Models (HMM) were designed. In [144], after ordering CCs in the text line using the horizontal coordinate of their centroid, the initial boundaries were estimated by the local minima of the horizontal projection. Then, CCs were grouped into sets based on the boundaries. A soft-margin SVM was adopted as a metric for separating between two adjacent sets. A global threshold was estimated by using unsupervised learning. In [159], the authors make use of the ground truth in terms of an ASCII transcription that is available in IAM

database. A HMM recognition system was used to automate, assigning each word in the ASCII ground truth the bounding box of the corresponding word. A statistical hypothesis testing method was presented by Haji et al [68]. The main idea was to learn the geometrical distribution of words within a sentence. A Markov chain and HMM were used. In [147] and [148], they tested five different supervised classification learning algorithms with different set(s) of features.

4.2.3 Historical Documents

Word segmentation was also applied to many types of historical documents. Historical document analysis is easier than handwritten document process if efficient binarization methods were applied on the manuscripts. Historical documents are written neatly in comparison to free handwriting. Manmatha et al. [100] presented a scale space approach. This method was based on blob creation that is based on CCs. In order to calculate the blobs, a differential expression based on second order partial Gaussian derivatives was used. An algorithm was introduced by Sanchez et al. [133] composed of three steps: (1) word candidate initialization using mathematical morphology; (2) merging dots and accents to the word box, and (3) punctuation mark splitting. In [60], a technique based on CCs labeling was introduced.

4.2.4 Word Segmentation Contests

Due to the importance of text segmentation, four Handwriting Segmentation Contests were organized: ICDAR2007 [63], ICDAR2009 [62], ICFHR2010 [61], and ICDAR2013 [145]. Therefore, a benchmarking dataset with an evaluation methodology were created to capture the methods' efficiency. The total number of participants on these competitions was thirty research groups with different algorithms. The results of the participated methods are given in Table 10.

The performance evaluation for the participated methods is based on counting the number of matches between the words detected by the algorithm and the words in the ground truth [127]. A MatchScore table was used whose values were calculated based on the intersection of the ON

pixel sets of the result and the ground truth. Based on a pixel approach, the MatchScore is defined as:

$$MatchScore(i,j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)}$$

where I is the set of all image points,

G_j is the set of all points inside the j ground truth region,

R_i is the set of all points inside the i result region,

T(s) is a function that counts the elements of set s,

Table MatchScore(i,j) represents the matching results of the j ground truth region and the i result region.

It is considered a one-to-one match only if the matching score is equal to or above a specified threshold. The detection rate (DR), recognition accuracy (RA) and performance metric (FM) are defined as follows:

$$DR = \frac{o2o}{N}$$

$$RA = \frac{o2o}{M}$$

$$FM = \frac{2 DR RA}{DR + RA}$$

where N is the count of ground truth elements,

M is the count of result elements,

o2o the number of one-to-one match.

Table 10: Results of the participated methods in segmentation contests in Latin scripts

Contest	Method	M	o2o	DR	RA	FM
ICDAR2007	BESUS	19091	9114	80.7%	52%	63.3%
	DUTH-ARLAS	16220	9100	80.2%	61.3%	69.5%
	ILSP-LWSeg	13027	11732	90.3%	92.4%	91.3%
	PARC	14965	10246	84.3%	72.8%	78.1%
	UoA-HT	13824	11794	91.7%	87.6%	89.6%
	RLSA	13792	9566	76.9%	74.0%	75.4%
	PROJECTIONS	17820	8048	69.2%	48.9%	57.3%
ICDAR2009	CASIA-MSTSeg	31421	25938	87.28%	82.55%	84.85%
	CMM	31197	27078	91.12%	86.80%	88.91%
	CUBS	31533	26631	89.62%	84.45%	86.96%
	ETS	30848	25720	86.55%	83.38%	84.93%
	ILSP-LWSeg-09	29962	28279	95.16%	94.38%	94.77%
	Jadavpur Univ	27596	23710	79.79%	85.92%	82.74 %
	LRDE	33006	26318	88.56%	79.74%	83.92%
	PAIS	30560	27288	91.83%	89.29%	90.54%
ICFHR2010	NifiSoft-a	15192	13796	91.18%	90.81%	91.00%
	NifiSoft-b	15145	13707	90.59%	90.51%	90.55%
	IRISA	14314	12911	85.33%	90.20%	87.70%
	CUBS	15012	13454	88.92%	89.62%	89.27%
	TEI	14667	13406	88.61%	91.40%	89.98%
	ILSP-a	14796	13642	90.17%	92.20%	91.17%
ICDAR2013	CUBS	23782	20668	87.86%	86.91%	87.38%
	GOLESTAN-a	23322	21093	89.66%	90.44%	90.05%
	GOLESTAN-b	23400	21077	89.59%	90.07%	89.83%
	INMC	22957	20745	88.18%	90.36%	89.26%
	LRDE	23473	20408	86.75%	86.94%	86.85%
	MSHK	21281	17863	75.93%	83.94%	79.73%
	NUS	22547	20533	87.28%	83.94%	89.13%
	QATAR-a	24966	20746	88.19%	83.10%	85.57%
	QATAR-b	25693	20688	87.94%	80.52%	84.07%
	NCSR (SoA)	22834	20774	88.31%	90.98%	89.62%
	ILSP (SoA)	23409	20686	87.93%	88.37%	88.15%
	TEI (SoA)	23259	20503	87.15%	88.15%	87.65 %

4.3 Our Method

In this stage, the procedure is divided into two steps. In the first step, the distance between adjacent components was computed using a gap metric while the second step deals with classifying the distances either as inter- or intra-word gaps. A writer dependent technique for estimating the threshold based on a given documents is considered more accurate than estimating the threshold for all the documents. In other words, spaces between words are part of a writing style, so writer dependent technique provides better result [94]. Thus, a global threshold across all documents is not a perfect solution; but in the case of IFN/ENIT database that is composed of images containing some words, we had to use a global threshold.

4.3.1 Distance Computation

In order to calculate the distance between adjacent components, we use geometrical features of the main components. The two gap metrics that are most used in the Latin-based language method of word segmentation are Bounding Box (BB) and Convex Hull (CH). The input to this stage is a binarized main component. We assume that each main component is either a word or part of a word, e.g. PAW, PAW fragment, character, or part of character. This means that the main component does not belong to more than one word. All the gaps between adjacent components (adjacent to their order) are measured to use them first to identify the threshold for inter- and intra-word gaps then they are used to classify each gap.

Bounding Box Metric

BB is composed of the smallest rectangle's coordinates within which all the points of the main component lie. After extracting the main components as shown in Figure 30(a), they are ordered from left-to-right. Then, a BB for each component is calculated, as shown in Figure 30(b). Then, all the overlapped bounding boxes (OBBs) are merged, Figure 30(c). OBB is defined as a set of CCs whose projection profile overlaps in vertical projection. To measure the distance between BBs, the minimum horizontal distances between pairs of adjacent BBs or \and BBs are used as seen in Figure 30(d).

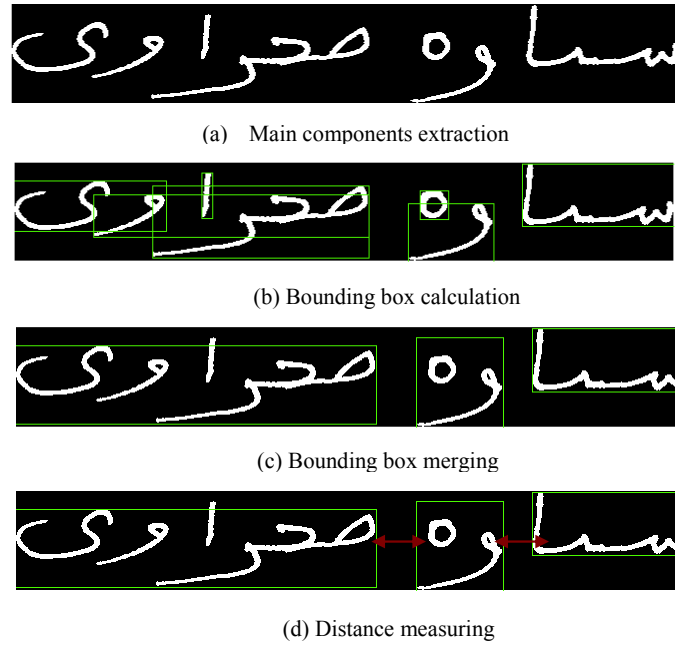


Figure 30: Steps illustration of metric-based segmentation using BBs

Convex Hull Metric

CH specifies the smallest convex polygon that contains the points of the main component. After extracting the main components, the CH of each components are generated in Figure 31(a), and the extreme points are extracted in Figure 31(b). The start and end points of each CH are identified by finding the shortest distance between extreme points. All the overlapped convex hulls (OCH) are combined in Figure 31(c). OCH occurs when intervals of start and end points overlap. To measure the distance between CHs, Euclidean distance is applied between adjacent CHs or/and OCHs in Figure 31(d). The Euclidean distance d between two points is the length of the line segment connecting them. The distance between j and q points is measured by:

$$d(j, q) = \sqrt{\sum_{i=1}^n (j_i - q_i)^2}$$

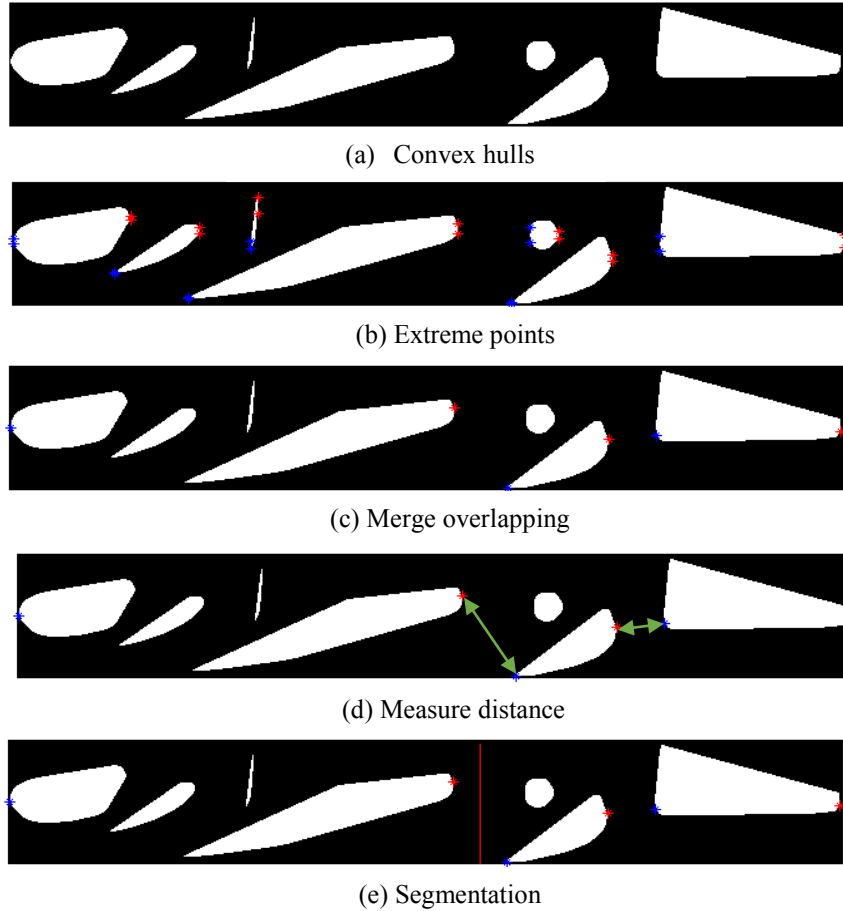


Figure 31: Steps illustration of metric-based segmentation using CHs

Baseline Dependent Metric

At this stage the distance between PAWs is calculated based on the baseline position. After estimating the baseline that is explained in Chapter 3, the CCs within the baseline range are extracted. Then, the BB of each extracted CC is calculated and the overlapping BBs are merged.

Next, the distance between adjacent BBs based on the baseline position is computed as shown in Figure 32.

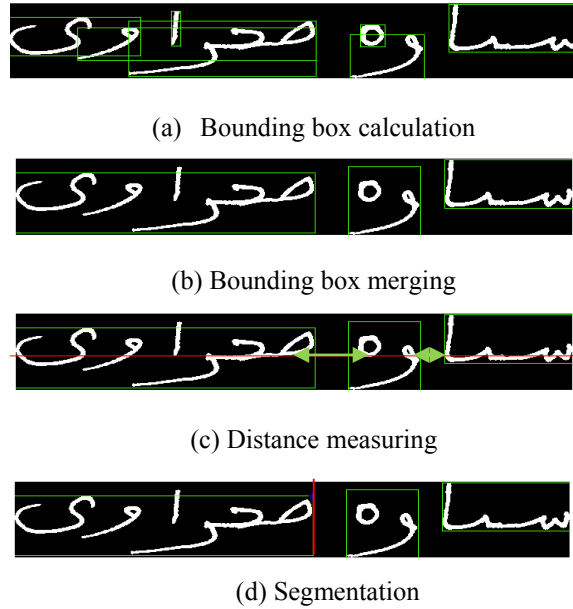


Figure 32: Steps illustration of metric-based segmentation using Baseline Dependent distance

4.3.2 Gap Classification

For a gap classification problem, we apply a novel approach used by G. Louloudis et al. [94]. This method is based on unsupervised learning of the already computed distances into two distinct classes that represent inter and intra word classes. They adapt the use of Gaussian Mixture Model, which is a type of clustering algorithm. A mixture model based clustering is based on the idea that each cluster is mathematically represented by a parametric distribution. There are two clusters problems, so every cluster is modeled with a Gaussian distribution.

To calculate the parameter for Gaussian Mixture Model, an iterative technique called Expectation Maximization is used. To start this technique, data points are selected randomly to be used as the initial means while the covariance matrix for each cluster is set to be equal to the covariance of the full training set. Each cluster is given equal prior probability. Expectation Maximization technique defines the clusters with two steps. In the first step, the probability that each data point p belongs to each cluster is calculated using the following equation:

$$w_j^{(p)} = \frac{g_j(x)\phi_j}{\sum_{s=1}^k g_s(x)\phi_s}$$

where $w_j^{(p)}$ is the probability that each data point p belongs to cluster j ,

$g_j(x)$ is the multivariate Gaussian for cluster j ; the probability of this Gaussian producing the input x ,

ϕ_j is the prior probability of cluster j ,

k is the number of cluster,

The equation for the probability density fraction of a multivariate Gaussian is

$$g_j(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-1/2(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)}$$

where j is the cluster number,

x is the input vector,

n is the input vector length,

Σ_j is the $n \times n$ covariance matrix of cluster j ,

$|\Sigma_j|$ is the determinant of the covariance matrix,

Σ_j^{-1} is the inverse of the covariance matrix.

In the second step, the cluster means and covariance based on the probabilities calculated previously are computed. In other words, it updates the model based on the previous calculated probabilities. The rules for the maximization step are:

$$\phi_j = \frac{1}{m} \sum_{i=1}^m w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^m w_j^{(i)} x^{(i)}}{\sum_{i=1}^m w_j^{(i)}}$$

$$\sum_j = \frac{\sum_{i=1}^m w_j^{(i)} (x^{(i)} - \mu_j) (x^{(i)} - \mu_j)^T}{\sum_{i=1}^m w_j^{(i)}}$$

The prior probability of cluster j , denoted as ϕ_j , is computed as the average probability a data point belongs to cluster j . The equation for μ_j is the average of all data points in the training set and each sample is weighted by the probability of belonging to cluster j . In the equation for the covariance matrix, each sample's contribution is also weighed by the probability that it belongs to cluster j . Finally, the thresholds for both inter- and intra-word gaps are determined based on each cluster's mean.

4.4 Experiments

We performed experiments using the IFN/ENIT [127] database. We applied our experiments on a subset from each set in the database. We used set-a for training and subsets from set-b, set-c, set-d, and set-e for testing. We evaluate the performance of our metric-based method manually since the ground truth of IFN/ENIT does not include any information on the word level. A tool is implemented to simplify evaluating the method manually. This tool outputs three results for each set; the accuracy of word segmentation, over and under segmentation errors. The accuracy is based on the correct number of extracted words. Over-segmentation occurs when a word is segmented to several parts. Under-segmentation occurs when more than one word are merged. Table 11, Table 12 and Table 13 show the word segmentation results stemming from the metric-based methods. Table 11 shows the result of the method that is based on BB, Table 12 shows the result of the method based on CH while Table 13 shows the result of the method based on baseline position. Table 13 shows only the result of set-d since the testing set of our baseline estimation method (Chapter 3) is set-d.

In the testing set of BB method, 765 out of 1229 words are extracted correctly. Therefore, the system performance is about 62.24 % on overall accuracy. Among error segments, we observe that under-segmentation error rate is higher than the over-segment error rate. The correct segmentation ranges from 67.34% to 83.12%. Set-e gets the best result in comparison to the other sets.

For the result of the metric-based method that uses the CH, the performance is lower than the method that based on BB. We believe that this low performance occurs because of the inconsistency of existing of ascender and descender in Arabic language. In contrast, in Latin-based language the use of CH achieved better result than BB. The accuracy of segmentation using CH for the four sets is 31.97 %. The accuracy for the four sets ranges from 24.03% to 46.48%. Over-segmentation error results when both BB and CH are almost the same; however, the under-segmentation error result of CH is much higher than the one for BB.

Our method that is based on the baseline position show a slightly higher performance than BB based method. The accuracy of segmentation using baseline position for set-d is 66.4%. Over segmentation is much lower than BB based method. We believe such low performance occur because of not considering (missing) descender parts. Comparison of these result will be presented in Chapter 7.

Table 11: BB results

Set	Total number of words	Correct segmentation		Over segmentation		Under segmentation	
		# words	%	# words	%	# words	%
Set-b	570	331	58.07	49	8.5	190	33.3
Set-c	227	164	64.31	10	4.4	53	23.3
Set-d	256	163	63.67	34	13.2	59	23
Set-e	176	107	60.79	4	2.2	65	36.9

Table 12: CH results

Set	Total number of words	Correct segmentation		Over segmentation		Under segmentation	
		# words	%	# words	%	# words	%
Set-b	570	137	24.03	41	7.9	392	68.7
Set-c	227	86	37.88	10	4.4	131	57.7
Set-d	256	119	46.48	21	8.2	116	45.3
Set-e	176	51	28.97	3	1.7	122	69.3

Table 13: Baseline dependent result

Set	Total number of words	Correct segmentation		Over segmentation		Under segmentation	
		# words	%	# words	%	# words	%
Set-d	256	170	66.4	75	29.29	11	4.29

4.5 Conclusions

In this chapter, we present a state of the art method through the comparison of two kinds of metrics, BB and CH. The use of BB outperforms the use of CH. Thus, we use BB in the first stage of our overall methodology. This method is based on calculating the distances between the CCs. The method is tested on a large number of images from IFN/ENIT database. Many errors are occurring as there is a lack of boundaries between words. The results of using CHs is much lower than using BBs, however this is not the case in Latin-based languages. We believe such a performance occur because of the large number of ascenders and descenders in Arabic writing. In addition, we introduced the use of a baseline dependent metric. However, we applied it only to set-d since the testing set for our learning-based baseline estimation is only applied on set-d. The result of using the baseline dependent metric showed better results for set-d in comparison to the performance of BB metric with the same set.

Chapter 5

Isolated Character Recognition System

Character recognition systems contribute to the field of automation process. Isolated character recognition is included in many systems such as office automation, postal address processing, business and data entry application. In this thesis, isolated character recognition is used to improve the performance of text segmentation that is explained in the next chapter.

In this chapter, we propose a standard recognition system with supervised learning. We apply some state-of-the-art techniques for recognition. Figure 33 shows a flowchart of the overall method of isolated character recognition system. A recognition system passes through three main stages that are discussed in this chapter: preprocessing, feature extraction and classification. In Section 5.1, we discuss some of the previous works of isolated Arabic character recognition. Then, we discuss the preprocessing stage, binarization, noise removal, size normalization and skeletonization in Section 5.2. The extracted features are described in Section 5.3. We applied Support Vector Machine (SVM) as a classifier with a Radial Basis Function (RBF) kernel that is described in Section 5.4. In Section 5.5, the database used is provided. The experiments and results are presented in Section 5.6. Finally, we summarize the chapter in Section 5.7.

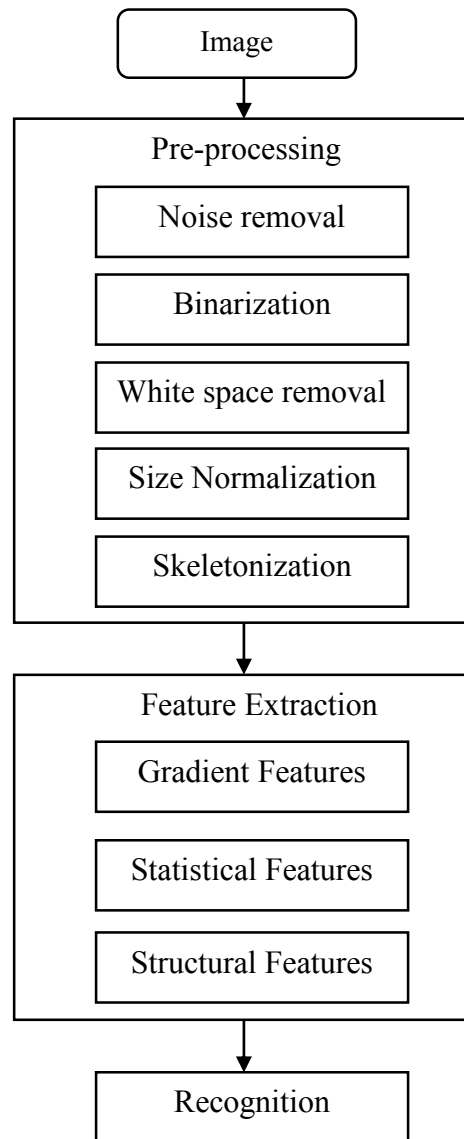


Figure 33: Isolated character recognition system

5.1 Previous Works

Previous works on recognition of isolated Arabic characters have been proposed. Some of them used only isolated shapes of the characters while others apply their methods on all the four forms of the characters (initial, middle, last and isolated). Various approaches have been introduced for online and printed systems, but our focus is solely on offline handwritten systems.

In [7], a system that recognizes the segmented handwritten character was presented. After skeletonizing the character, it was converted into a tree structure. A set of fuzzy constrained character graph models was designed. These graph models with fuzzily labeled arcs were used as prototypes. Some rules were applied to match the character tree to the graph model.

Sahlol et al. extracted several features from both main and secondary components of the characters [132], and [131]. The systems used K-Nearest Neighbor and Support Vector Machine for character recognition. The experiments have been conducted on CENPARMI's isolated character database [20]. The recognition results were 82.5% and 89.2 % respectively.

Neural network have been used in many approaches. A discrete Wavelet transform was utilized for Farsi and Arabic character classification [108]. Features were extracted using Haar Wavelet. A forward Neural Network using Backpropagation were used for recognition. In [24], a Neural Network combined with structural features was applied by extracting features from tracing skeletons. Discrete Cosine Transform and Discrete Wavelet Transform were used for feature extraction [89]. Artificial Neural Network is used in the classification stage. Dehghan et al. [46] utilize Zernike, Pseudo Zernike and Legendre Moments for feature extraction. An unsupervised learning was applied using neural network.

In fact, several methods were based on the characters' skeleton. In [3], 96 common features were extracted from the main component, secondary component, character skeleton and boundary. Then, five different feature selection techniques were applied to choose among the features. A structural method with statistical features was proposed in [158]. After thinning the characters, the skeleton was segmented into primitives. These primitives were categorized into loops, curves or line segments. In addition, some feature points were extracted. The Nearest Neighbor Multi-Layer Perceptron with Euclidean distance were used for classification. In [123], each character was represented by polygonal based on the extracted contour. The character models were built while utilizing the directions and length features of the polygonal approximation. Fuzzy logic and turning

angle functions were used in the classification phase. Khedher et al. [85] used approximate stroke sequence matching.

In addition, some techniques extracted the structural features. Five classifiers have been tested in [64] with different types of features. The best result obtained was with Linear Discriminant Analysis. In [84], two types of features were extracted, called qualitative and quantitative features. Quantitative features focus on character height, width, area, and dot numbers. Meanwhile, qualitative features include branches, dot positions, connection points, and loops. The recognition approach is based on feature matching. In [154], different features were extracted and heuristics were applied. Several features have been extracted and analyzed [2]. Features were extracted from both the whole image and from the partitioned image [80].

Different techniques were also proposed for the recognition of isolated characters. In [1], the moment features were extracted from the whole character, the main component and the secondary components. A multi-objective genetic algorithm was applied to select the most efficient feature subsets. Aburas et al. [8] proposed an algorithm for new construction of optical character recognition system similar to wavelet compression technique. This algorithm is based on that the wavelet compressed image is a decomposition vector that can uniquely represent the input image to be correctly reconstructed later at decompression phase. A. Amin [25] proposed a conventional method that relied on hand-constructed dictionaries. The inductive learning was used and based on first-order Horn clauses. In [110], the fractal code was used for feature extraction and Multilayer perceptron neural network was applied for recognition. Fractal code and wavelet transform were used for extracting features while the support vector machine was used for classification [111], and [109].

Features were extracted from built graph [77]. Dictionary matching was used for recognition. In [34] a statistical approach was applied. The main and secondary components were isolated and recognized separately. The moments of the horizontal and vertical projections of the main components were calculated. For classification stage, quadratic discriminant functions were applied.

5.2 Preprocessing

Preprocessing is a very essential phase in any recognition system since the final result relies on this stage. In fact, there are many different tasks for preprocessing and their effects vary depending on the data type. In general, preprocessing includes noise removal, skew correction, slant detection, baseline estimation, representation and normalization. Our preprocessing stage is composed of noise removal, binarization, white space removal, size normalization, and skeletonization.

Noise Removal

Noise is usually caused by faulty memory locations in hardware, transmission in a noisy channel or multifunctioning pixels in camera sensors. In general, it is imperative to remove corrupted pixels to facilitate many image processing tasks. For Arabic text, noise removal, or more precisely salt and pepper noise removal, is indispensable, since dots in Arabic language have a huge effect on recognition and these types of noise can reduce the systems's performance.

We apply Median Filtering to remove the noise of the images. The aim of filtering is to diminish spurious points as salt and pepper noise. Median Filter is a nonlinear process that preserves edges while removing random noise. The main idea is to convolve a predefined mask to assign a value to the centered pixel base on its neighborhood pixels. In other words, the output pixel is set to median of the neighborhood pixel values. We adopt a window of size 3×3 . In Figure 34, the image before and after noise removal is illustrated.

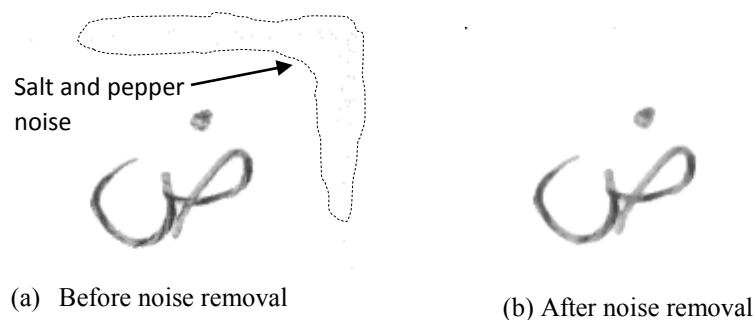


Figure 34: Noise removal

Binarization

In image processing, it is essential to select an adequate threshold to extract text from their background. We use Otsu's [118] method to find the threshold to minimize the intra class variance of the black and white pixels. This method assumes that the image contains two classes of pixels called foreground and background. The optimum threshold separating those two classes is calculated so that their combined spread variance is minimal. Figure 35 shows the character image before and after binarization.

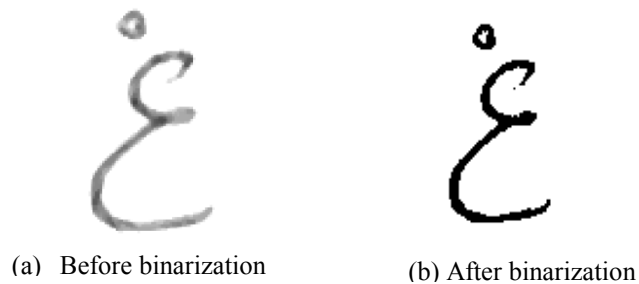


Figure 35: Binarization process

White Space Removal

White space around the character in an image does not help in the recognition stage, so this space is removed. We use bounding box to remove the white space. The smallest rectangle containing the pixels of the character is located. Then, the white space outside the rectangle is eliminated as shown in Figure 36.

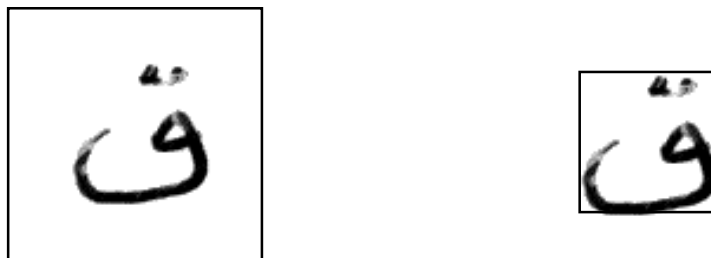


Figure 36: White space removal

Size Normalization

Before feature extraction, size normalization is considered as an important phase. Each image is normalized into two different sizes, 64 x 64 pixels and 128 x 128 pixels, using an aspect ratio adaptive normalization strategy [130]. Two different sizes of the image are used for different feature extraction processes.

Skeletonization

For some feature extraction process, the character must be standardized for both the training and testing phases. Thus, we apply skeletonization algorithm using Zhang-Suen thinning method [162] as explained in Chapter 7. This algorithm removes all the contour pixels except those belonging to the skeleton, Figure 37.

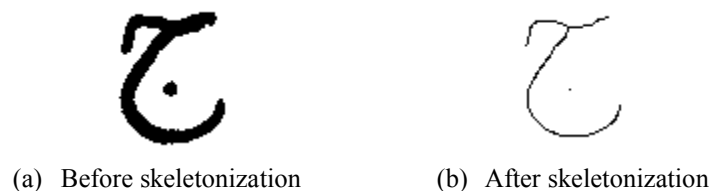


Figure 37: Skeletonization

5.3 Feature Extraction

Classifiers cannot efficiently process the raw images on their own. Thus, feature extraction is an important process that aims at reducing the dimensionality of an input data and extract useful information. Selection of salient features is one of the primary decisions in designing a recognition system to achieve high performance. The features' classes in the literature on handwritten recognition are: a) structural features are intuitive aspects of writing, such as loops, writing baseline, ascenders, or turning points, b) statistical features are numerical measures that is language independent, and c) a combination of both. Then, those features are converted into vectors that are used to calculate a score, which can be either probability or distance, for matching the input image with possible interpretation. In this chapter, we investigate many statistical and

structural features by using supervised learning for Arabic character recognition. We extracted gradient features and structural features. Several experiments were conducted with different features to find the best combination of these features that produces the best results.

Extraction of Gradient Features

One set of features that we extract is the gradient feature. Gradient features maintain both the position and the direction information of the image. In the gradient feature extraction phase, as explained in [130], each image of size 128 x 128 pixels is converted into a grayscale image. Robert's filter masks were applied on the images. These masks are shown in Figure 38, below:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Figure 38: Robert's Filter Mask for Extracting Gradient Features

Let $IM(x, y)$ be an input image; the horizontal gradient component (g_x) and vertical gradient component (g_y) were calculated as follows:

$$g_x = IM(x + 1, y) - IM(x, y)$$

$$g_y = IM(x, y + 1) - IM(x, y)$$

- The gradient strength and direction of each pixel $IM(x,y)$ were calculated as follows:

$$\text{Strength: } s(x, y) = \sqrt{g_x^2 + g_y^2}$$

$$\text{Direction: } \theta(x, y) = \tan^{-1}(g_y / g_x)$$

Some examples of gradient images are shown in Figure 39, below:

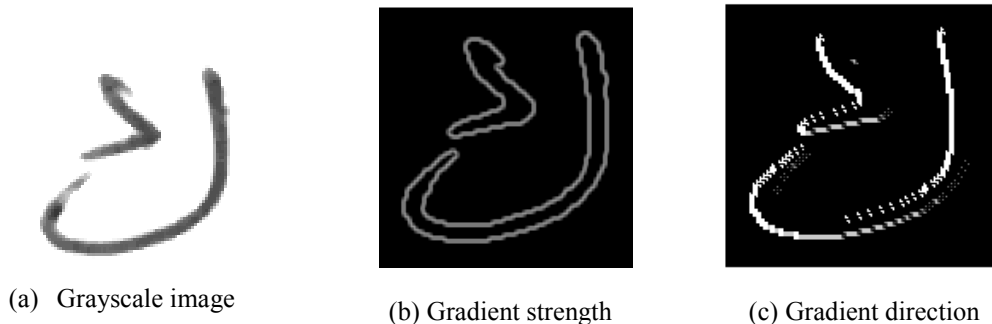


Figure 39: Gradient features

After calculating the gradient strength and direction for each pixel, the following steps were taken in order to calculate the feature vector:

1. The direction of a vector (g_x, g_y) in the range of $[\pi, -\pi]$. These gradient directions were quantized to 32 intervals of $\pi/16$ each.
2. The gradient image is divided into 81 blocks, with 9 vertical blocks and 9 horizontal blocks. For each block, the gradient strength is accumulated in 32 directions. By applying this step, the total size of the feature set in the feature vector is $(9 \times 9 \times 32) = 2592$.
3. To reduce the size of a feature vector, a 5×5 Gaussian filter is applied by down sampling the number of blocks from 9×9 to 5×5 . The number of directions is reduced from 32 to 16 by down sampling the weight vector $[1 \ 4 \ 6 \ 4 \ 1]$. The size of the feature vector is 400 (5 horizontal blocks \times 5 vertical blocks \times 16 directions).
4. A variable transformation ($y = x^{0.4}$) is applied on all features to make the distribution of features Gaussian-like.

Extraction of Statistical Features

In addition to the gradient feature, other statistical features are extracted. They are Horizontal Projection (HP) which is explained in Section 4.7, along with Vertical Projection (VP) profiles which provides the number of black pixels in each column. Moreover, the number of black pixels for the whole image is calculated.

Extraction of Structural Features

Some structural features are extracted, such as the end, and intersection points. For point's extraction, we use the hit-or-miss transform which is the tool used for shape detection. The transform of a set of x by a structuring element $A = (A_1, A_2)$ is a set of points x such that when the origin of A coincides with x , A_1 fits x and A_2 fits the complement of x :

$$HMT_A(x) = \{ x | (A_1)_x \subseteq X \text{ and } (A_2)_x \subseteq X^c \}$$

Moreover, the upper profile features were used to capture the outline shape of the top part [13]. To extract the upper profile feature, the following steps were followed:

- Each image is converted into a two-dimensional array.

- For each column, the vertical distance is measured from the top of the image to the closest black pixel by counting the number of white pixels.

Feature vector

After extracting the gradient and structural features from each image, all the features were merged to make a feature vector size of 468 (400 gradient features, 64 upper profiles, 4 structural features). Then, this feature vector is used in the classification phase.

5.4 Recognition

A Support Vector Machine (SVM) is a technique in the field of statistical learning. SVMs have shown to provide good results for both offline and online cursive handwriting recognition [161]. In addition, SVM outperformed several other classifiers in [160]. An SVM maps the input space non-linearly into a very high dimensional feature space. The aim of SVM is to separate the hyperplane optimally with maximal margin. This approach is based on the training data points that are located at the margin, called support vectors.

Given a set of labeled training data $(y_1, x_1), \dots, (y_l, x_l)$, where $y_i \in \{-1, 1\}$ and $x_i \in \mathbb{R}^n$, an SVM tends to solve the following optimization problem:

$$\min_{w,b,\varepsilon} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \varepsilon_i$$

$$\text{subject to} \quad y_i (w^T \phi(x_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0$$

where ϕ maps the training vectors x_i to a higher dimensional space,

ε_i are slack variables that permit margin failure,

C is the parameter that trades off wide margins with a small number of margin failure.

We use an open source library for the implementation of SVM called LibSVM [42]. The input of LibSVM is a feature matrix and the output is the classification result probabilities. LibSVM uses a Radial Basis Function (RBF) kernel for mapping a nonlinear sample into a higher sample space. RBF is given by:

$$K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}$$

where x_i is the support vector,
 x_j is the testing data point,
 γ is the kernel parameter.

In our experiment, these optimal parameters are chosen by using v-fold cross validation via parallel grid search on the validation set. A training model is generated for the whole images' collection with their class labels.

5.5 Database

An offline Arabic handwritten isolated letter dataset was developed by CENPARMI in 2008 [20]. The dataset contains 36 isolated letters. The data were collected in Saudi Arabia and Canada from 328 Arabic writers of different nationalities, ages, genders, and educational levels or background. This database is composed of 12693, 4367, and 4366 training, validation, and testing samples respectively.

5.6 Experiments

Several experiments have been conducted on the above Isolated Arabic handwritten letters dataset. For the first experiment, we use all the classes of the basic Arabic characters which consist of 28 classes. Table 14 shows the recognition rates, which is the correct classification over the total number of samples, of using different features. The comparisons between our method with M.T. Sahlol et al [132], and [131] conducted on the same database are provided in Table 15. Our results were much lower compared to M.T. Sahlol. However, most of the misclassifications were due to the confusion between the classes that have the same main shape but with only different diacritics. An example of such confusion is given in Figure 40. Such a problem can be solved with adding weights to diacritics feature vectors or using multiple classifiers, one of them for diacritics' classification.

Since our main objective is to recognize the main shapes of the characters, we perform an experiment using only these shapes of the character as in [108]. We combine the character's classes

that have the same main shape as shown in Table 16 and there are 14 classes. We perform several experiments to obtain the best results as shown in Table 17. In addition, since our concern is with only endWord and non-endWord classes which is a two-class problem, we perform an experiment based on these two classes, endWord and non-endWord classes. The recognition rate of this experiment is 99.49%.

Table 14: Results of our method using all the classes of Arabic characters

Feature vector	Gradient feature	Horizontal projection	Vertical projection	Upper profile	Structural features	Size of feature vector	Recognition rate
Fv1	x					416	72.24%
Fv2	x	x				480	71.91%
Fv3	x	x	X			480	73.52%
Fv4	x	x	X		x	420	72.27%
Fv5	x		X		x	464	72.63%
Fv6	x		X	x	x	468	75.85%

Table 15: Comparison between different methods

Method	Recognition rate
[131]	88%
[132]	89.2%
Our method	75.85%



Figure 40: Confusion between classes

Table 16: Combined classes

Class	Characters
01	آ ا ا ا
02	ب ت ث ن
03	ج ح خ
04	د ذ ر ز
05	س ش ص ض
06	ط ظ
07	ع غ
08	ف ق
09	ك
10	ل
11	م
12	هـ
13	و و
14	ى ي ئ

Table 17: Experimental results with different features

Feature vector	Gradient feature	Horizontal projection	Vertical projection	Upper profile	Structural features	Size of feature vector	Recognition rate
Fv1	x					416	89.97%
Fv2	x	x				480	89.66%
Fv3	x		X			480	89.51%
Fv4	x				x	420	90.27%
Fv5	x			x		464	90.73%
Fv6	x			x	x	468	90.88%

5.7 Conclusions

In this chapter, we study the previous work of Arabic handwritten isolated character recognition. Different sets of commonly used features were tested on isolated characters. In addition, several experiments were conducted with different numbers of classes. The combination of Gradient features, upper profile, and structural features outperform the other extracted features.

Chapter 6

CENPARMI Arabic Database for Handwriting Recognition

One of the most important aspects in this thesis is the final shape of the character, either connected or isolated. Since many final shapes are not available in the isolated characters dataset, we choose to create a new database. This database contains word images that are composed of the shapes of all Arabic characters at all positions (beginning, middle and final). This chapter represents the work towards developing a new database for offline Arabic handwriting recognition. Section 6.1 summarizes the existing Arabic databases. The data forms are explained in Section 6.2. In Section 6.3, the process of extracting the items from the forms is described. Then, an overview of the database is given in Section 6.4. In Section 6.5 the ground truth is explained. Finally, the chapter is concluded in Section 6.6.

6.1 Related Works

Described in Section 1.8 and used for this thesis, the IFN/ENIT [127] is the most widely used database in the Arabic research field. Al ISRA database [79] is composed of sentences, words, signature, and digits gathered from 500 students. The database, created by the Linguistic Data Consortium, comprises 9693 handwritten pages [146]. The Centre for Pattern Recognition and Machine Intelligence (CENPARMI) has developed two Arabic databases. In 2003, Al-Ohali et al. [14] created a database for Arabic cheques that includes legal and courtesy amounts. In 2008, Alamri et al. [20] developed a database containing digits, dates, numerical strings, words, letters and some symbols. AHDB [13] is a database for words in legal amounts and it also contains handwritten pages of 100 writers. Khedher et al. [85] developed a database of unconstrained Arabic handwritten characters. ADBase is a database of handwritten Arabic digits (Indian) [52]. KHATT is the most recent database [99] which consists of 1000 handwritten forms written by 1000 distinct writers. In addition, it covers all of the Arabic character shapes. However, we are not aware of any database that contains all the shapes of the characters in different positions which are separated into classes. Such database can be extremely beneficial for the research of Arabic handwritten texts since it can determine the problems of segmentation and recognition more precisely. Moreover, this database can facilitate different types of experiments.

6.2 Data Collection

The form includes 63 words and 10 digits. We tried to find the minimum number of words that include all the shapes of the characters in all positions. Table 18 shows all the words that are used for each character in each position. Some of them are not in the same position as shown in the Table but it is written in the same way. We do not include the isolated characters. We design a specific data-entry form to collect these words and digits from Arabic native speakers. One of the forms written by one writer is shown in Figure 41. The data is written in light colored boxes to facilitate data extraction. In addition, four black boxes were added to the corners of the forms to use their coordinates for skew correction, if needed, and to locate target area. We requested some personal information from the writers like their gender, age, and whether they are right-handed or

Table 18: Letter shapes

Isolated	Beginning	Middle	End
ا	الحبيب	أمال	أمال
أ	ألم	مرفاً	مرفاً
إ	إحسان	الإسلام	--
آ	أمال	مأل	مأل
ب	بخار	عنكبوت	ذئب
ت	تل	ممتلى	كبريت
ث	ثمرة	مثلث	مثلث
ج	وجه	فجر	ثلج
ح	إحسان	ضحى	صحيح
خ	خليج	بخار	فخ
د	دمشق	وليد	وليد
ذ	ذئب	بذرة	بذرة
ر	زراعة	طاهر	طاهر
ز	زراعة	مزن	مزن
س	سلك	مسؤول	شمس
ش	شمس	دمشق	عش
ص	صحراء	بصر	شخص
ض	ضحى	أخضر	بيض
ط	طاهر	معطوف	نفظ
ظ	ظفر	مظهر	لفظ
ع	علف	ملعقة	مطلع
غ	غيم	بغل	صانع
ف	فجر	شفق	سيف
ق	قمر	ملعقة	شفق
ك	كبريت	عنب	سلك
ل	لفظ	ثلج	بغل
م	محمد	شمس	ألم
ن	نحاس	عنكبوت	لبن
هـ	هذه	مظهر	هذه وجه
و	وجه	عنكبوت	عنكبوت
ي	كبريت	سيف	قاضي
ة	--	--	الأجهزة ملعقة
ى	--	--	ضحى
ء		ولاء	ولاء
أ	الأجهزة	مأ	مأ
إ	الإسلام		
لا	ولاء	الإسلام	الإسلام
ؤ	--	مسؤول	مسؤول
ئ	ذئب	بنر	ممتلى
	الحبيب	الخبز	المحنة

ARASHP0051

الرجاء عدم الكتابة في هذه المنطقة عربي	Arabic Handwritten Word Collection Form CENPARMI, Concordia University, Canada Email: am_jamal@cenparmi.concordia.ca	Address: 1455 de Maisonneuve Blvd. West, Suite EV003.403, Montréal, Québec, Canada, H3G 1M8 http://www.cenparmi.concordia.ca
اليد المستخدمة: اليد اليمنى <input checked="" type="checkbox"/> اليد اليسرى <input type="checkbox"/>	الجنس: ذكر <input type="checkbox"/> أنثى <input checked="" type="checkbox"/>	الفئة العمرية: من ١٠ إلى ٢٠ <input checked="" type="checkbox"/> من ٢١ إلى ٣٠ <input type="checkbox"/> من ٣١ إلى ٤٠ <input type="checkbox"/> من ٤١ فما فوق <input type="checkbox"/>
- فضلاً ضع إشارة في المربع المناسب		
- الرجاء كتابة كل كلمة في المكان المخصص لها و الإلتزام بحدود المربع مع مراعاة وضوح الخط		

محمد	بئر	الإسلام	مرفأ	ممتلى
محمد	بئر	الإسلام	مرفأ	ممتلى
صحراء	إحسان	نحاس	تل	كبريت
صحراء	إحسان	نحاس	تل	كبريت
مثلث	ثمرة	فجر	زراعة	ضحى
مثلث	ثمرة	فجر	زراعة	ضحى
ثلج	بخار	خليج	فخ	ألم
ثلج	بخار	خليج	فخ	ألم
عنب	لبن	وجه	هذه	وليد
عنب	لبن	وجه	هذه	وليد
آمال	قاضي	مدار	ولاء	ذنب
آمال	قاضي	مدار	ولاء	ذنب
بذرة	مآل	مأ	الأجهزة	مزن
بذرة	مآل	مأ	الأجهزة	مزن
سيف	شمس	دمشق	عش	بصر
سيف	شمس	دمشق	عش	بصر
شخص	أخضر	بيض	طاهر	مطلع
شخص	أخضر	بيض	طاهر	مطلع
نفت	ظفر	مظهر	لفظ	معطوف
نفت	ظفر	مظهر	لفظ	معطوف
غيم	بغل	صانغ	علف	قمر
غيم	بغل	صانغ	علف	قمر
ملعقة	شفق	عنكبوت	سلك	مسؤول
ملعقة	شفق	عنكبوت	سلك	مسؤول
الحبيب	الخبز	المحنة	٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢ ١ ٠	
الحبيب	الخبز	المحنة	٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢ ١ ٠	

Figure 41: Filled form

left-handed. Even though all this personal information is not used in this research, it may be significant for other researchers [10]. The forms have been filled in by participants in both cities Makkah and Jeddah in Saudi Arabia. We gathered the handwriting samples from 650 writers. Participants were asked to write the samples within the box boundaries using a dark pen.

6.3 Data Extraction

After the forms were filled, digital versions were obtained. We save those images with a resolution of 300 Dots per Inch (DPI) and 24 bit color depth. The data extraction process started with the removal of red boxes from the forms. After extracting all the samples, a special filter is applied to remove salt and pepper noise on each image. Besides the true color images, the forms are saved in both gray-scale and binary. The coordinates of the area for each handwritten element on the true color form are located manually. After the red boxes are removed from the true color forms, a special program is developed to automate the data extraction process. By taking each box's coordinates, this program extracted the box image in its true color. The images from the same box of each form are saved in a unique folder. Once the databases were created in true color images, they are converted into greyscale and binary formats. The binary format is used for letters extraction. We develop a program to extract the letters manually.

6.4 Database Overview



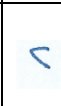







Three datasets are created for the word images: true color, gray-scale, and binary formats. The forms are stored in uncompressed TIFF-file format. The data is divided into training, validation and testing sets with the approximate distributions of 60%, 20%, and 20% respectively. The total number of word images is 24,570 used for training, 8,190 for validation and 8,190 for testing. This means 390 writers are used for training and the rest for validation and testing. Statistics of this word dataset are given in Table 19. Every word image is saved with a name and a number indicating its writer. For example, the word image is saved as "ARASHP0001_W001.TIFF", where ARASHP0001 refers to the form's number, and W001 identify the word's number in the

form. For ESL dataset, we used the binary word images for letter segmentation. Each segmented final shape letter is saved in a new image with a form’s number, word’s number, and a letter code with its position. In addition, we gather Arabic digits (Indian) for future works. Each form contains 10 digits, as shown in Table 20.

Table 19: Statistics of letter shapes

Number of writers	Total number of images	Training set	Validation set	Testing set
650	47,450	28,470	9,490	9,490

Table 20: Handwritten Indian Digits

0	1	2	3	4	5	6	7	8	9
									

6.5 Ground Truth

In the final structure of our database, each folder, containing handwritten samples, contains the ground truth data file for these samples. The ground truth data file includes some information about each sample: image name, content, number of CCs, writer’s number, writer’s age, writer’s gender, and writer’s handwriting orientation (left-handed or right-handed). An example of the ground truth data for Arabic letter shape dataset is given in Table 21.

6.6 Conclusions

This new database can help overcome new challenges in the area of Arabic document analysis and recognition. We use the end shape of each letter to improve the segmentation of Arabic texts. For future work, all the word images can be segmented into letters in all positions.

Table 21: Example of the ground truth data for Arabic letter shape dataset

Image Name	ARASHP0001_W001_TahE01.TIFF
Content	آ
Writer No.	ARASHP0001
Item No.	W001
Gender	F
Hand Orientation	R

Chapter 7

End Shape Letter Recognition-Based Segmentation

Our main contribution is discussed in this chapter. As mentioned before, due to the lack of differences between the inter- and intra-word gaps, we study the structure of Arabic language. One of the most distinguishable characteristics is that most of the letters in the Arabic alphabet have different shapes based on the letter's position within the word. Many letters have different shapes at the end of the words. We utilize this fact to facilitate segmenting the texts into words. In Section 7.1, we discuss the work related to word segmentation for the Arabic language. An overview of our approach and the rationale behind its use, is described in Section 7.2. The details of our algorithm are given in Section 7.3. The experiment is explained in Section 7.4. In Section 7.5, the error analysis of the text segmentation is given. Section 7.6 presents time complexity of the system, while the comparison of the results with Arabic text segmentation is provided in Section 7.7. Finally, we conclude our work on this aspects in Section 7.8.

7.1 Related Works

Word segmentation is a critical step towards word spotting and text recognition. Many word segmentation techniques can be found in the literature. Nevertheless, it is still a challenging problem in handwritten documents considering there is little research on Arabic handwritten texts segmentation. Some works used manual segmentation as part of their methodology [69] to apply their methods.

In [29], an online Arabic segmentation method was proposed. The gap types are classified based on local and global online features. The fusion of multi-classification decisions was used as a post-processing stage to verify the decisions. The method were applied on the sentence dataset are collected for online handwriting research. This dataset is collected on tablet PCs from 48 participants and is composed of 154 documents.

J. Alkhateeb et al. proposed a method for Arabic handwritten texts segmentation into words based on the distances between PAWs and the words [22]. Vertical projection analysis was employed to calculate the distances while the statistical distribution was used to find the optimal threshold. Bayesian criteria of minimum classification error were used to determine the threshold. The technique was applied on a subset of the IFN/ENIT database. The correct segmentation of one-word and three-word images was 80.34% and 66.67% respectively.

In [23], an offline handwritten Arabic texts segmentation technique was introduced. First, the CCs of the images were detected based on the baseline. Their bounding boxes were determined. These boxes were extended to include the dots and any small CCs. The distances between adjacent PAWs were obtained. They assumed that the distance between words is larger than the distance between PAWs. Based on that assumption, a threshold approach was used. Two conditional probabilities were determined by manually analyzing more than 200 images. A Bayesian histogram minimum classification error criterion was used to find the optimal distance.

M. Kchaou et al applied scaling space to segment Arabic handwritten documents into words [78]. The techniques that were used for segmentation were scaling space and feature extraction from horizontal and vertical profiles. Two documents written by five writers were used in their experiments. Segmentation errors varied between 29.5% and 3.5%. They believe that the errors arise from different writer styles, coordinating conjunctions and distances between PAWs.

Srihari et al. [142] proposed a segmentation method by extracting several features on both sides of a potential segmentation points using a neural network. The process starts with extracting CCs. Main and secondary components are merged into groups. Nine features were extracted from those groups. The features are: the distance between BB of adjacent groups, widths of adjacent groups, character ‘Alif’, minimum distance between CHs, and the ratio between the sums of the areas enclosed by the CHs of the individual group to the total area inside the CH enclosing the clusters together. The correct segmentation rate is 60% over 10 writers each writing 10 documents. The results of these Arabic word segmentation methods is given in Table 22.

Table 22: Results of Arabic word segmentation method

Method	No. Images	Image Type	System Type	Method Type	Result
[22]	106	IFN/ENIT	Offline	Threshold	66-91%
[23]	200	IFN/ENIT	Offline	Threshold	85%
[142]	100	Document	Offline	Classification	60%
[78]	5	Document	Offline	Scaling	71.5-97.5%
[53]	154	Document	Online	Classification	82.69

7.2 ESL-Based Segmentation

To distinguish our segmentation approach from previous methods, we utilize the knowledge of Arabic writing by recognizing the last letter of PAWs. Some authors pointed out the importance of using the language specific knowledge for Arabic texts segmentation [22], [53], [130] and [23]. Our approach for segmentation is a two-stage strategy: (1) metric-based segmentation (discussed previously in Chapter 4), and (2) ESL based segmentation.

As known, the Arabic alphabet has twenty-eight letters. Twenty-two letters have different shapes when they are written at the end of a word as opposed to the beginning or in the middle. In addition, two extra letters, which are not part of the alphabet, have different shapes at the end of a

word. Therefore, recognizing these shapes can help identify the end of a word. Fourteen main shapes can be used to find the segment points. The remaining letters have the same main part but with different number and/or position of dots. Only NLC letter shapes are written the same way at the beginning, the middle or the end of a word. Therefore NLC letters cannot be used to identify the end of a word. Consequently, ESLs can be categorized into two classes: endWord and non-endWord. Figure 42 shows some samples of endWord class letters. In this stage, the main idea is to recognize the ESL that helps to specify the word segmentation points. ESL can be isolated or connected as part of a PAW. However, the end-shape needs to be detected first before recognition can begin. Each step is described in the following sections. Our method is depicted in Figure 43.

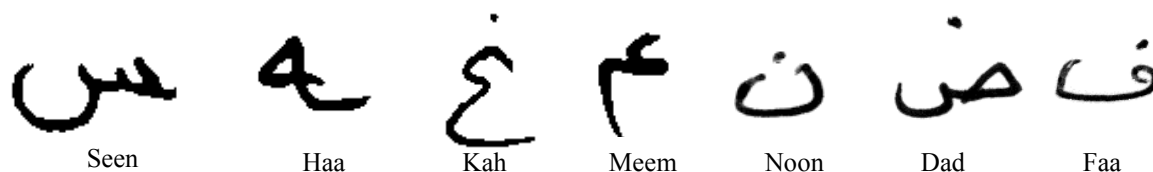


Figure 42: Some samples of endWord class letters

7.3 Our Proposed Algorithm for Text Segmentation

This section presents an automatic segmentation of Arabic texts into words using ESL-based approach. A binary image of the text is the input to the algorithm. The main steps of the algorithm are given in Figure 43.

Extracting and Labeling Connected Components

The Connected Components (CCs), consist of connected black pixels of a text image, were extracted. The foreground pixels are negated by being assigned a value of 1 while the background pixels are assigned with 0. Normally, a PAW is composed of several CCs: a main component, and secondary components such as diacritics, and/or directional markings. Therefore, the first main step in segmenting an Arabic handwritten word is detecting and labeling its CCs. CC analysis is the most efficient approach since the Arabic script consists of several overlapping CCs. The eight-connected neighboring pixels method is used. At this step, the PAW is extracted from the image including both main and secondary components. Then, the secondary components are removed as

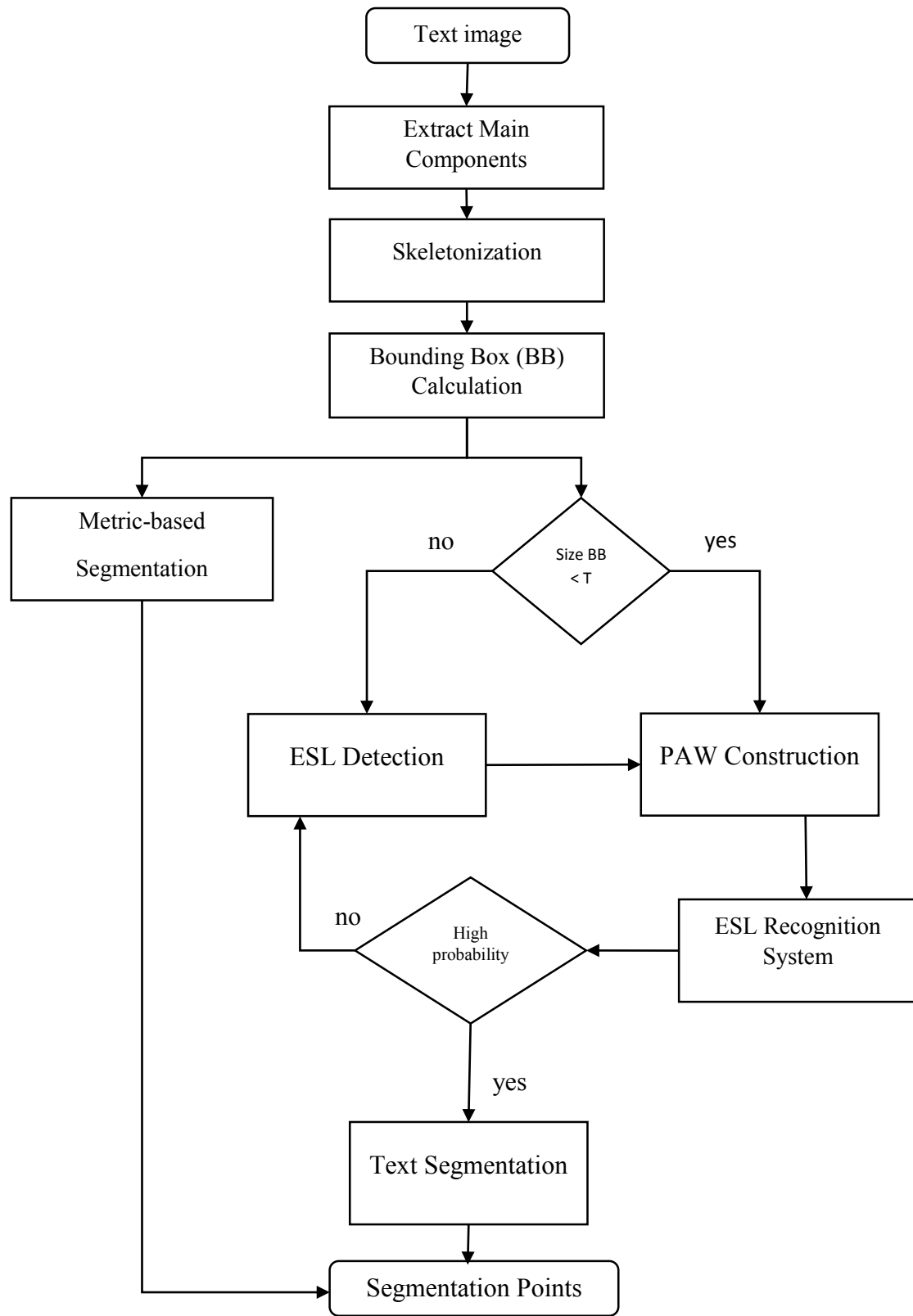


Figure 43: Block diagram of the proposed method

explained in Chapter 2, and saved in a new image. Next, the main components of PAWs are labeled from left to right.

Skeletonization

We apply thinning to facilitate both extracting the starting points and some important features for segmentation and tracing the PAW. The Zhang-Suen thinning algorithm [162] takes the Boolean image and reduces it to an 8-connected skeleton. In this algorithm, each iteration consists of two steps. Four points are removed in the first step: the southeast boundary points, northwest corner points, northwest boundary points, and southeast corner points. Let p_1, p_2, \dots, p_8 represent eight neighbour pixels of point p . Let $B(p)$ be the number of nonzero eight neighbour pixels of pixel p and let $A(p)$ be the number of zero-to-one conversions in the ordered sequence p_1, p_2, \dots, p_8 . If p is a contour point, then the following four conditions should be satisfied in order to flag the pixel p for removal:

1. $2 \leq B(p) \leq 6$
2. $A(p) = 1$
3. $p_1 \cdot p_3 \cdot p_5 = 0$
4. $p_3 \cdot p_5 \cdot p_7 = 0$

The removal of pixel p is delayed until all the pixels of the image are examined by the algorithm. In the second step, the contour point p is flagged for removal if the following four conditions are satisfied:

1. $2 \leq B(p) \leq 6$
2. $A(p) = 1$
3. $p_1 \cdot p_3 \cdot p_7 = 0$
4. $p_1 \cdot p_5 \cdot p_7 = 0$

After applying the two steps to all the pixels of the image, the resulting image is the skeleton.

Bounding Box of Main Components

The height and width of each main component is computed. This task is used for two purposes. The first is for identifying the isolated letter to avoid segmentation of the letter while the second is for applying metric-based method to calculate the distance between the BBs between adjacent main components to find the segmentation path between words.

ESL Detection

At this stage, the main purpose is to detect the last letter of a PAW. If the bounding box of the main component is bigger than a threshold that is found to be the best value for isolated letter tested in CENPARMI isolated letters training set, then the main component is considered as PAW that contains more than one letter and thus the PAW must be segmented.

1. **Find the start point**

The starting point is determined by searching the PAW's image from left-to-right and finding the first black pixel and its coordinates.

2. **Trace the main component**

The skeletonized main component is traced. The row and column coordinates of the black pixels of the main component are saved in a two element vector.

3. **Determine the segmentation points**

The main component is traced until a change occurs in the traced component by finding the local minima or local maxima depending on the starting points if it is an ascender or descender.

4. **PAW Construction**

Based on the coordinates of the segmented main component, all the secondary components above and below this component are extracted. A PAW image is constructed by combining the main and secondary components, and saved in a new image.

5. **Recognition**

After extracting the isolated letter or determining the segmentation point, the PAW image is constructed and passed to ESL recognition system. If the recognition probability of the segmented PAW is low, the third step is repeated in order to find a new segment point.

PAW Construction

If the BB's size is less than a threshold T , the isolated letter (PAW) is constructed based on the coordinates of the isolated main component. The threshold are based on studying a large number of isolated letters samples. An isolated letter is constructed by combining the main and secondary components.

ESL Recognition

At this stage, either the detected ESL of the PAW or the isolated letter is sent to an ESL recognizer. This recognizer classifies the end-shapes of the PAWs and isolated letters. We use both the CENPARMI Arabic isolated letter database and our new set (ESLs) while the SVM is used for classification. We used parts of both ESLs and isolated letters to accelerate the training and testing processes.

Segmentation

We consider two approaches for segmentation: 1) metric-based, and 2) ESL-based. The metric-based technique is based on the distance between two adjacent bounding boxes of the main components. The ESL recognition method depends on the class of the detected letter and its probability.

Algorithm

Given a text image T , let $\{mc\}$ be the sequence of all the extracted main components from T , such that for any two main connected components mc_i and mc_{i+1} , mc_i is on the left of mc_{i+1} . Let $\{sc\}$ be the sequence of the extracted secondary components from T , for $j = 1 \dots n$. In addition, the bounding box is calculated for each mc_i and denoted as bb_i . The output is a list of segmentation points.

$mc \leftarrow \text{extracted main components}(T)$

$sc \leftarrow \text{extracted secondary components}(T)$

```

for all  $mc_i \in mc$  and all  $bb_i \in bb$  do

    if  $bb_i < Threshold$  then
         $PAW \leftarrow ConstructPAW(mc_i, sc_j)$ 
         $[Class, Probability] \leftarrow Recognize(PAW)$ 
    else
        find starting point
        repeat
             $Trace(mc_i)$ 
            find local minima or local maxima
             $SMC \leftarrow Segment(mc_i)$ 
             $PAW \leftarrow ConstructPAW(SMC, sc_j)$ 
             $[Class, Probability] \leftarrow Recognize(PAW)$ 
        until Probability is high
    end if

     $Distance \leftarrow DistanceBetweenBB(bb_i, bb_{i+1})$ 
     $DetermineSegmentationPoints(Distance, Class, Probability)$ 
end for

```

7.4 Experiments

We perform several experiments using the IFN/ENIT [127] database. These experiments have been conducted to test different sets of features, sets of training samples and sets of thresholds. Figure 44 illustrates the steps of our algorithm.

An initial experiment is conducted to test the main idea of our approach. The first experiment conducted is based on Arabic isolated letters so that it can be used to find the segment points. As mentioned before, the letters used as identification of segmentation can be in two forms: isolated or connected (part of PAW). If it is part of PAW, detection of ESL is needed for recognition. We use only the CENPARMI Arabic isolated letters database which is a subset of the ESL.

ربايع سيدى ظاهىر

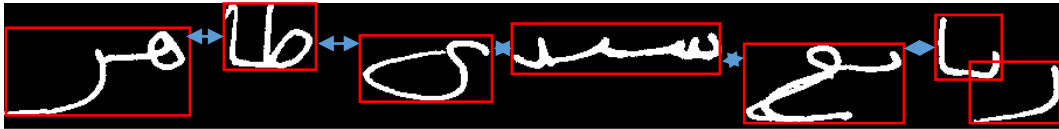
(a) Input image



(b) Main components extraction



(c) Bounding Box calculation



(d) Distance calculation



(e) ESL segmentation



(f) PAWs Construction



(g) Text segmentation

Figure 44: Steps of text segmentation algorithm

The CENPARMI isolated Arabic letters database is composed of only the isolated shapes of the letters (which is a subset of the final shape letters). In addition, there is no available database that has all the shapes of the letters in their final positions that are connected. Thus, we use only CENPARMI database in this preliminary experiment. We used the IFN/ENIT images that can be segmented with the isolated letters and are composed of two and three words [74]. This set contains a total of 440 unique names. We used set-a for training and subsets from set-b, set-c, set-d, and set-e for testing. The total numbers of training and testing samples are 50 and 285 respectively. Table 23 shows the word segmentation results of metric-based and ESL-based methods.

Table 23: Results of metric-based and ESL-based methods using subsets of IFN/ENIT

Set	# images	Metric-based	ESL-based
Set-b	50	67.34%	82.00%
Set-c	82	68.29%	84.15%
Set-d	77	83.12%	85.61%
Set-e	76	71.05%	86.84%

The second experiments are done on all the IFN/ENIT images that contain two to four words. Set-a is used for training while set-b, set-c, set-d, and set-e are used for testing. In this experiment, the ESL database, which is described in Chapter 6, is used. This database is composed of both isolated and connected end shape letters. The total number of images for testing is 998. The results of our method are given in Table 24.

Table 24: Results of our algorithm of text segmentation algorithm

Set	Total images	# images	number words	Correct segmentation		Over segmentation		Under segmentation	
				#words	%	# words	%	# word	%
Set-b	329	100	223	196	87.89	13	5.8	14	6.26
		200	460	412	89.37	14	3.04	34	7.39
		300	688	624	83.2	17	2.47	47	6.83
		329	771	596	85.3	20	2.5	155	20.1
Set-c	230	100	220	194	88.18	6	2.7	20	9.09
		200	429	371	86.48	18	4.19	40	9.32
		230	491	420	85.53	20	4.07	51	10.38
Set-d	244	100	222	196	88.28	4	1.8	22	9.9
		200	445	375	84.27	14	3.15	56	12.58
		244	535	421	78.69	17	3.17	97	18.13
Set-e	186	100	203	163	80.3	12	5.91	28	13.79
		186	379	307	80.1	29	7.72	43	11.34

7.5 Error analysis

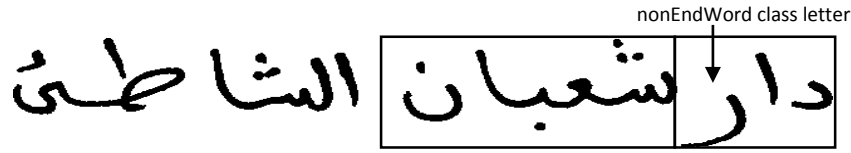
Two types of errors are produced by our system: under and over segmentation. Each type of error is caused by a number of reasons. We describe below the causes of each type.

Under Segmentation:

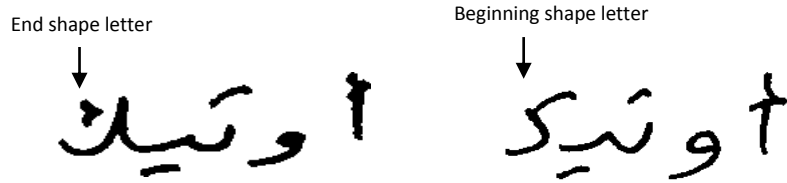
- When the distance between bounding boxes is small and ESL belongs to non-endWord class. (Figure 45.a)
- When endWord class letter is written like beginning or middle letter, or unfamiliar shapes. Our ESL datasets does not contain such shapes (Figure 45.b)
- When completely overlapping components do not belong to any PAW, and PAW construction error occurs. (Figure 45.c)
- Error of segmenting ESL. (Figure 45.d)

Over Segmentation:

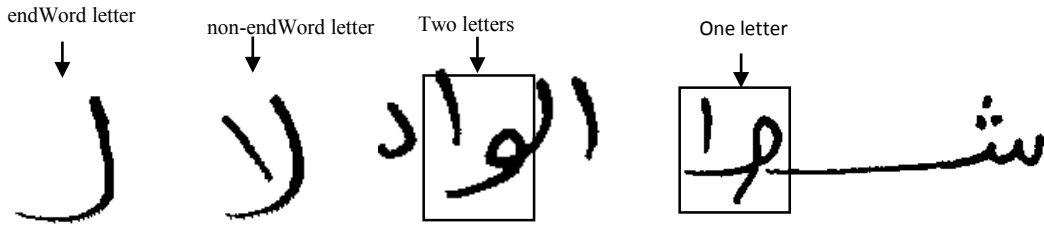
- Some non-endWord class letters are written similarly to endWord class ones due to writing styles. (Figure 45.e)
- When a PAW is broken, confusion between classes occurs. (Figure 45.f)
- Unclear images. (Figure 45.g)



(a) Short distance and nonEndWord class letter



(b) Short distance and non-endWord class letter



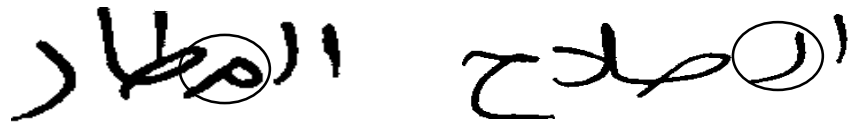
(c) PAWs construction error



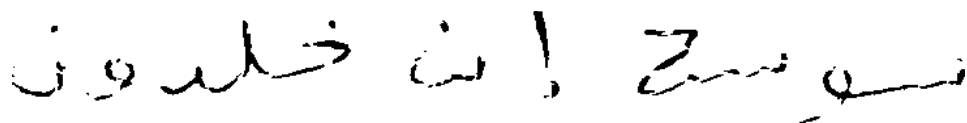
(d) Segmentation error



(e) Confusion between classes



(f) Broken PAW



(g) Unclear images

Figure 45: Some common sources of errors

7.6 Time Complexity

All experiments were conducted on a system with 8.0 GB RAM, Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz 2.20GHz, Windows 8.1 OS, 64-bit operating system, x64-based processor and MATLAB code. The system under study is divided into four main procedures; namely secondary components removal, baseline estimation which includes baseline dependent feature extraction, metric-based segmentation that includes metric calculation and ESL-based segmentation which includes ESL detection and recognition. Table 25 shows the time complexity for the proposed system (time in seconds). The time complexity is calculated per segmentation point decision for both metric-based and ESL-based segmentation methods for randomly selected images. The results show that the metric-based approach operates at several times faster than the speed of the ESL-based approach. For the ESL-based approach, most of the time is consumed by segmentation and classification.

Table 25: Time complexity of the systems in seconds

Image	Metric-based	ESL-based
Image#1	0.00013	60.796477
Image#2	0.000127	44.61412
Image#3	0.00024	58.289716

7.7 Comparison of Results with Arabic Text Segmentation

To enable consistent evaluation, we compare our Arabic handwritten text segmentation with the system that uses the same database, and applied on 200 randomly selected images from set-d that are composed of one to four words. The two systems presented in [22] and [23], with which we compare our system, are based on a threshold approach. In [22], vertical projection analysis was applied to calculate the distances between PAWs and words. To find the optimal threshold, Bayesian criteria of minimum classification error was used. In [23], the distances between

bounding boxes was employed. The threshold was determined by manually analyzing more than 200 images. However, the training set was not determined. In addition, these approaches conducted their tests on the images that are composed of one and more than one words. The comparison between our algorithm with these two methods is provided in Table 26.

Table 26: Comparisons of methods of Arabic handwritten text segmentation

Method	Number of images	Correct segmentation rate
[23]	200	85%
[22]	106	79.66%
Our method	200	89.65%

7.8 Conclusions

In this chapter, we introduce a methodology for segmenting handwritten texts into distinct words. The main novelty of the proposed approach is to utilize the knowledge of Arabic writing. This method is based on recognizing the ESL of the PAWs which help to find the segment points. After calculating the width of the bounding box of each PAW, the need of PAW segmentation is determined. If the width does not exceed the threshold, this means that the PAW is an isolated letter. On the other hand, if the width is more than a threshold, the ESL detection process begins until the ESL is recognized. Based on the recognition result of the ESL and the metric-based distance, as explained in Chapter 4, the segmentation point is identified. From our experimental results, it is shown that the proposed method outperforms existing methods for Arabic text segmentation conducted previously on the IFN/ENIT database. Most of the errors are caused by either NLC letters or the broken PAWs.

Chapter 8

Arabic Handwritten Word Recognition

Generally, in a handwriting recognition process, the word is segmented into characters and then the classifier recognizes each character. However, character segmentation is not simple, especially in Arabic systems which have to confront many obstacles. Hence, researchers avoid letter segmentation and choose to apply segmentation-free methods or holistic approaches. In fact, Word recognition can be divided into word-based and character-based. Word-based recognition systems aim at identifying the word from its global shape without either explicit or implicit segmentation, and it is called segmentation-free. Character-based approaches can be categorized into segmentation-based and recognition-based. In the former, the input image is partitioned into sub-images, which can be classified. In the latter, unit segmentation is a by-product of recognition process and the segmentation and recognition are accomplished simultaneously. In this chapter, we focus on holistic approaches where the word is modeled as a whole. The aim of this chapter is to implement a holistic word recognizer to facilitate word spotting which is discussed in Chapter 9. In Section 8.1, we discuss some proposed approaches for holistic Arabic word recognition. Our method is described in Section 8.2 while the experiments that we conducted using different databases are presented in Section 8.3. Finally, we conclude this chapter in Section, 8.4.

8.1 Related Works

Holistic approaches can be divided into two categories: implicit segmentation and segmentation-free [66]. The disadvantage of using this approach is that the lexicon should be limited. On the other hand, the advantage is that they bypass the character segmentation problem. We review only segmentation-free approaches applied to Arabic and Farsi scripts in this section.

Several holistic approaches were applied on Arabic printed text. In [92], the word shape is analyzed with a vector of morphological features. This vector is matched against a database which is a precomputed vectors from a lexicon of Arabic words. A. Amin used machine learning to generate a decision tree for classification [27]. In [140] and [86] each word is transferred into a normalized polar image. Then, a two dimensional Fourier transform was applied. The recognition is based on the Euclidean distance.

Few holistic methods have been proposed for offline handwritten words recognition. Three of those methods used Farsi whose alphabet was derived from Arabic. These methods used a database that consists of about 17,000 images of 198 Iranian city names. In [47], the word recognition system is based on fuzzy vector quantization and HMM. HMM was trained by modified Baum-Welch algorithm and they achieved 67.19% recognition rate. M. Dehghan et al [48] proposed an approach using a discrete HMM and Kohonen self-organizing vector quantization. The histogram of chain code direction is used as feature vectors. A 65% correct recognition rate was achieved. In addition, B. Vaseghi et al. [163] used a discrete HMM and self-organizing vector quantization with some preprocessing tasks such as binarization and noise removal. The correct recognition rate was 66.42%.

In [12], CENPARMI Arabic word and IFN/ENIT databases were used to test a holistic word recognition system. Several sets of features were tested, including Gradient features, Gabor filter features and Frequency features using Discrete Fourier transform. Moreover, two statistical classifiers, Modified Quadratic Discriminant Function (MQDF) and Regularized Discriminant Analysis (RDA), and one discriminant classifier, Support Vector Machine (SVM) were examined.

8.2 Database

An offline database of Arabic handwritten words was developed by CENPARMI in 2008 [20]. The database contains 69 words. The dataset includes weights, measurements, and currencies. The data were collected in Saudi Arabia and Canada from 328 Arabic writers of different nationalities, ages, genders, and educational levels. This database is composed of 17007, 4485, and 4233 training, validation, and testing samples respectively.

8.3 Word Recognition System

Our handwritten word recognition system is similar to our isolated letter recognition system described in Chapter 5, with slight modifications.

8.3.1 Preprocessing

In the preprocessing stage, we followed the steps that are developed for isolated letter recognition: noise removal, binarization, white space removal, size normalization, and skeletonization. In addition, the recognition system is examined using slant correction method. However, many methods have been introduced for slant correction and applied on the IFN/ENIT database; however, they either did not provide the effect of slant correction on recognition, or had a slight improvement [132], [114], [157], and [112]. We used the method proposed in [131].

8.3.2 Feature Extraction

For the feature extraction phase, in addition to the gradient feature that is computed by Robert operator, and explained in Chapter 5, we also test the gradient feature that is computed by Sobel operator which has two masks for calculating the horizontal and vertical gradient components that are applied in [130]. Figure 46 shows the Sobel masks for extracting gradient features.

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Figure 46: Sobel masks for extracting gradient features

Let $I(u,v)$ an input image, the horizontal gradient component (g_x) and vertical gradient component (g_y) are computed as follows:

$$g_x = I(u - 1, v + 1) + 2I(u, v + 1) + I(u + 1, v + 1) - I(u - 1, v - 1) - 2I(u, v - 1) - I(u + 1, v - 1)$$

$$g_y = I(u - 1, v - 1) + 2I(u - 1, v) + I(u - 1, v + 1) - I(u + 1, v - 1) - 2I(u + 1, v) - I(u + 1, v + 1)$$

In addition to the structural features that are explained in Chapter 5, more features are extracted to improve the performance. These features include the number of holes, number of connected components, and number of end points based on the baseline position.

8.3.3 Recognition

The two optimal parameters are chosen by using v -fold cross validation. The total number of feature points is 599 (400 gradient features, 64 horizontal projection, 64 vertical projection, 64 upper profile, and 7 structural features).

8.4 Experiments and Results

In our experiments, we extracted the gradient features by using the Sobel filter to compare our results with the Robert filter. There is a slight difference in the results and the performance with Robert filter is better than Sobel. The comparison in experimental results of both filters is shown in Table 27.

Table 27: Sobel filter vs. Robert filter

Image size	Robert Filter	Sobel Filter	Upper Profile	Results
128 X 128	X			85.79%
	X		X	87.58%
		X		83.48%
		X	X	83.48%
64 X 64		X		83.76%
	X			86.12%
	X		X	86.77%
		X	X	83.77%

Moreover, several experiments are performed to evaluate different sets of features. Table 28 shows the results of using horizontal and vertical projection, gradient features, upper profile, and structural features. In addition, several features are extracted based on baseline position.

Table 28: Comparison of recognition results with different features

	Gradient Feature	Upper Profile	Horizontal Projection	Vertical Projection	Structural Features	Based on Baseline	Recognition results
F1	X	X					87.58
F2	X	X	X				87.77
F3	X	X		X			87.58
F4	X	X	X	X			87.41
F5	X	X			X		95.39
F6	X	X			X	X	97.86

8.5 Comparison with Arabic word recognition system

We compare our system with a holistic based system that used Modified Quadratic Discriminant Function with gradient features [12]. This method was applied on CENPARMI Arabic handwritten words. Moreover, it was also conducted using IFN/ENIT in which sets a, b, c, and d used for training and set-e for testing. To enable consistent performance evaluation, we perform our experiments on the same databases. Table 29 shows the recognition rates by applying these two holistic methods to both CENPARMI and IFN/ENIT databases. The approaches that are applied on CENPARMI have high recognition rates since it has a relatively small lexicon (69 word classes). On the other hand, the methods that are applied on IFN/ENIT database have performed poorly because it has a large lexicon (937 word classes).

Table 29: Recognition results using CENPARMI and IFN/ENIT database

Method	CENPARMI	IFN/ENIT
[12]	96.89	55.54
Our system	97.86	65.11

8.6 Conclusions

Different sets of features are tested on CENPARMI Arabic handwritten words database with a lexicon of 69 word classes. The classifier is based on holistic approach. In addition, several size normalization were tested with Robert and Sobel filters.

Chapter 9

Impact of Text Segmentation on Word Spotting

Word spotting, also called searching or indexing, is the task of detecting words within a document, and it is an effective way for document retrieval. Several studies favor word spotting over word recognition for word retrieval. Word spotting gained significant attention among researchers. Many papers have addressed word spotting based on Latin and Chinese documents. However, few studies were proposed for Arabic documents since the nature of Arabic writing is more challenging. Our aim of this chapter is to study the impact of text segmentation on word spotting. In Section 9.1, we discuss word spotting systems for the Arabic language. Performance evaluation metrics of word spotting systems is given in Section 9.2. Our method of Arabic word spotting system is described in Section 9.3, while the experiments are provided in Section 9.4. The error analysis is discussed in Section 9.5. Finally, the chapter is summarized in Section 9.6.

9.1 Word Spotting in the Arabic Language

Arabic handwritten word spotting systems can be categorized into word-based, PAW-based, and character-based. Clustering or segmenting the documents into words is considered the first task in many word spotting systems [12]. However, in Arabic scripts, the boundaries between words are often non-existent and arbitrary. Thus, only one research attempted to automatically segment the Arabic texts into words for a word spotting system [51]. However, they achieved 60% segmentation accuracy while another method segmented the texts manually into words [142].

Many methods proposed a script independent word spotting system [143], [153], [128], and [91]. However, they faced problems with words having the same root. Moreover, they found that low performance resulted when applying the system on the Arabic script.

Since there are no clear boundaries between Arabic words in handwritten documents, many methods apply PAW-based approaches. In [134], the PAW is converted into Word Shape Tokens (WST) and each PAW was represented by global structural features. Saabni and El-Sana [160] used Dynamic Time Wrapping (DTW) and HMM for matching PAWs. Khayyat et al. [80] proposed a learning based word spotting system using PAW-model with hierarchical classifiers. While in [82], they used a combination of PAW and word models with an ensemble of classifiers. In [83], PAWs are recognized separately, then PAW language models were used with the classifier to reconstruct words. In [141], a word matching system was implemented by extracting contour features from PAWs. Then, each PAW was embedded into Euclidean space while active-DTW was used to find the final matching result.

In addition to PAW-based approaches, segmentation-free methods are applied. Some segmentation-free approaches were proposed for Arabic handwritten word spotting, in which the segmentation process is embedded within the classification stage. These systems are either based on the character-model [153], or an over segmentation [44].

Word spotting systems have been applied to historical documents. In [106], Euclidean distance enhanced by rotation was applied with DTW to measure the similarity between two PAWs. E. Syakol et al. [136] and S.A. Shahab et al. [138] proposed a content-based method using a codebook. Several features were extracted to represent codes of PAW, characters, or symbols. Then distance measure, or similarity matching, was applied to find the final match.

However, we believe correct pre-segmentation of the texts into words will improve both system speed and performance. Improvement in the speed is achieved by eliminating some pre-processing tasks such as generating the words, clustering... etc. Improving the performance is a result of including more features (information and outcomes) from the whole word instead of PAWs.

9.2 Performance Evaluation

To evaluate the performance of a word spotting system, two metrics are used: recall and precision rates. Recall Rate (RR) measures the ratio of the successful retrieval of the word sample, or the actual positives. Precision Rate (PR) is the probability that the retrieved image is the target word. These metrics are calculated with the following formulas:

$$RR = \frac{TP}{TP + FN}$$

where TP (True Positive) is the total number of correctly spotted words, and

FN (False Negative) is the total number of the target words that are not spotted.

$$PR = \frac{TP}{TP + FP}$$

where FP (False Positive) is the total number of the spotted words that are misrecognized.

The precision-recall curve is used to calculate the Mean Average Precision (MAP) represented by the area under the curve.

9.3 Method Implemented

Our word spotting system uses the holistic handwritten word recognition system for recognizing the words. After segmenting the text into words, the word spotting system is applied. We apply this system to explore the benefit of text segmentation. Our system tries to recognize the target

words and reject all the other words with low probability. The extracted features of the target (template) image are passed to the system.

As mentioned earlier, LibSVM generates the classification probability of the tested samples. On the basis of these probabilities, the system outputs the classes of the samples. First, the system extracts the feature vector. Then, it maps this feature vector into feature space. In the feature space, the system finds the closest hyperplane of Class N. If the class of a sample is not trained, which means it does not belong to the target words, SVM maps that sample into the closest class.

In our system, after text segmentation into words, each image goes through the following process, as discussed in Chapter 8:

- Size normalization,
- Gradient features,
- Upper profile features,
- Structural features,
- Statistical features,
- Labeling feature vector,
- The feature vector is sent to holistic handwriting word classifier, and
- The result is returned in probabilities, we assume this word sample belongs to the class with the highest probability. However, if the probability is below a defined threshold, this sample is considered as false positive error.

9.4 Experimentation

To initially evaluate the impact of text segmentation on a word spotting system, two CENPARMI handwritten documents were manually segmented into words. The result of the spotting system is shown in Table 30. We compared our result with the system that was proposed by M. Al-Khayat et al. who used the CENPARMI database. Table 31 shows the results of the systems. Based on the results, we can see that the word spotting system after correct text segmentation was able to recognize a higher number of target words (86.66% vs. 50%-53%).

Table 30: Result of word spotting system after manual segmentation

Total Words	Total Targeted Words	True Positive TP	True Negative TN	False Positive FP	False Negative FN
99	15	13	66	20	2

Table 31: Comparisons of results on Word spotting

Method	Model	Precision	Recall
[80]	PAW	84.56%	50%
[83]	PAW	65%	53%
[82]	Word/PAW	84.4%	50%
Manual experiment	Word	39.39%	86.66%

We test the word spotting system on the IFN/ENIT database since it is the database that used for text segmentation. The text images with different names in IFN/ENIT are mixed. In other words, they are not separated into different classes. We collected all the images that have common words in the same folders. However, the IFN/ENIT is a huge database, it does not have many training samples. We only found three classes with more than 50 training samples. So, we manually segment three words from 189 images to create three classes. Each class is composed of more than 40 samples. These samples are segmented from sets a, b, c, and d. Then, these classes are trained using an SVM classifier. After text segmentation, word spotting system is applied. Set-e is used for word spotting system. The result is given in Table 32.

Table 32: Result of word spotting system on IFN/ENIT database

Total Words	Total Targeted Words	Total candidate words	True Positive TP	True Negative TN	False Positive FP	False Negative FN
379	42	368	35	255	73	5

9.5 Conclusions

In this chapter, we explore the benefit of text segmentation by applying word spotting system. We achieved 32.4% precision rate and 87.5% recall rate on the IFN/ENIT database. Most of the errors occur because of segmentation error. The word spotting accuracy can be improved by working on the segmentation error.

Chapter 10

Conclusions and Future Works

The purpose of this thesis is to design and implement an ESL-based text segmentation system that can overcome lack of the boundary problem which appears in Arabic handwritten texts. Several systems for Arabic handwritten texts are proposed and implemented in this thesis, and the results are presented.

10.1 Concluding Remarks

Automatic processing of Arabic handwritten texts is a challenging task. We use the knowledge of Arabic language to facilitate text segmentation. In order to achieve satisfactory performance, we have designed and implemented several systems. These sub-systems incorporate the process essential for an accurate secondary component removal (Chapter 2), learning-based baseline estimation (Chapter 3), metric-based segmentation (Chapter 4), ESL recognition system (Chapter 5), and ESL-based segmentation (Chapter 7). To explore the benefit of text segmentation, we implement two systems for word spotting: word recognition system (Chapter 8) along with word spotting system (Chapter 9).

Our method of secondary component removal is based on reconstructing the image on a generated mask. This method facilitates both the learning-based baseline estimation approach by

removing unneeded pixels, and metric-based segmentation technique by reducing the number of calculated BBs. We achieved 2.90% of false positives and 1.75% of false negatives.

For our learning-based baseline estimation method, we use some state of the art preprocessing algorithms to extract several baseline dependent features. These features include horizontal projection, centroid of convex hulls and average of horizontal line segments. The method shows great result with images containing only one word. Moreover, the performance is better when the number of training samples is higher. It reaches 96.27% for ≤ 5 error in pixels.

The metric-based segmentation approach, also called threshold-based method, is one of the common techniques that are used for text segmentation into words. We use two well-known metrics called bounding box and convex hull. For determining the threshold, Gaussian Mixture Models is used. Furthermore, we introduce baseline dependent metric that outperforms both the methods that applied bounding box and convex hull on set-d.

ESL-based segmentation is proposed to solve the lack of boundary that occurs in Arabic handwritten texts. This method aims at extracting the last letter of PAWs and recognizing it since some Arabic letters identify the end of a word. To enable ESL-based segmentation method, a classifier is implemented for recognizing ESLs. In addition, a handwritten word recognition system, which is a holistic approach such that the segmentation of the words into letters is not required, is used for word spotting.

10.2 Future Works

The following sections suggest improvements that can be made to enhance the performance of each step involving Arabic handwritten texts segmentation and recognition. The results achieved in this thesis encourage further research in several areas.

Databases

As part of the future work, the algorithms will be applied to other databases such as KHATT. More written ESLs need to be extracted in order to improve the recognition rate of ESLs to deal with a variety of handwriting styles. The system described in this thesis is applied on a database written by 411 writers. For recognition system, two datasets are used, one was written by 360 participants and the other was written by 650 participants. Increasing the database's samples will improve training the system since it will include different handwriting styles which allow more features to be developed.

Pre-processing

For the removal of secondary components, some diacritics based on the size need to be removed since some of them are near the mask. For baseline estimation method, we need a larger database to apply our method on one word image that can reach better results than the images that contain more than one word. In addition, more baseline dependent features need to be extracted. We will study the effect of slant correction on the images.

Feature extraction

More features can be added and tested. Adding different types of features may improve the recognition results that lead to better segmentation results. Moreover, feature selection needs to be examined to accelerate the recognition system.

Classification

The classification can be improved by testing different classifiers and by using hierarchical or multiple classifiers. Using separate classifiers for diacritics may avoid the confusion that occurs with letters that have the same main part with a different number or positions of diacritics. Also, applying weights on some features can avoid such confusion.

Segmentation

In our system, improving ESL-based segmentation and recognition performance can improve text segmentation performance. Experiments with different gap metrics as well as different threshold types may yield significant improvement.

References

1. Abandah, G. and Anssari, N. , Novel moment features extraction for recognizing handwritten Arabic letters. *Journal of Computer Science*, 2009. **5**(3): pp. 226-232.
2. Abandah, G. A. and Khedher, M. Z., Analysis of handwritten Arabic letters using selected feature extraction techniques. *International Journal of Computer Processing of Languages*, 2009. **22**(01): pp. 49-73.
3. Abandah, G. A. and Malas, T. M., Feature selection for recognizing handwritten Arabic letters. *Dirasat Journal Engineering Sciences*, 2010. **37**(2).
4. Abu-Ain, T., Abdullah, S. N. H. S., Bataineh, B., Abu-Ain, W. and Omar, K., Text normalization framework for handwritten cursive languages by detection and straightness the writing baseline. *Procedia Technology*, 2013. **11**(1): pp. 666-671.
5. Abu-Ain, T., Abdullah, S. N. H. S., Bataineh, B., Omar, K. and Abu-Ein, A., A Novel Baseline Detection Method of Handwritten Arabic-Script Documents Based on Sub-Words, *In Soft Computing Applications and Intelligent Systems*, S. Noah, et al., Editors. 2013, Springer Berlin Heidelberg. pp. 67-77.
6. Abu-Ain, T., Sheikh Abdullah, S. N. H., Omar, K., Abu-Ein, A., Bataineh, B. and Abu-Ain, W., Text Normalization Method for Arabic Handwritten Script. *Journal of ICT Research & Applications*, 2013. **7**(2):pp. 164-175.
7. Abuhaiba, I. S. I., Mahmoud, S. A. and R. J. Green, Recognition of handwritten cursive Arabic characters. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 1994. **16**(6): pp. 664-672.
8. Aburas, A. A. and Rehiel, S. M. A., Off-line omni-style handwriting Arabic character recognition system based on wavelet compression. *Arab Research Institute in Sciences & Engineering*, 2007. **3**(4): pp. 123-135.
9. Adiguzel, H., Sahin, E. and Duygulu, P., A Hybrid for Line Segmentation in Handwritten Documents. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012. pp. 503-508
10. Al Maadeed, S. and Hassaine, A., Automatic prediction of age, gender and nationality in offline handwriting. *EURASIP Journal on Image and Video Processing*, 2014. **2014**(1): pp. 1-10.
11. Al-Dmour, A. and Fraij, F. ,Segmenting Arabic Handwritten Documents into Text lines and Words. *International Journal of Advancements in Computing Technology*, 2014. **6**(3):pp. 109-119.

12. Al-Khayat, M., Learning-Based Arabic Word Spotting Using a Hierarchical Classifier. 2014, Ph.D Thesis, Department of Computer Science and Software Engineering, Concordia University.
13. Al-Ma'adeed, S. A. S., Recognition of off-line handwritten Arabic words. 2004, Ph.D Thesis, University of Nottingham.
14. Al-Ohali, Y., Cheriet, M. and Suen, C. Y., Databases for recognition of handwritten Arabic cheques. *Pattern Recognition*, 2003. **36**(1): pp. 111-121.
15. Al-Rashaideh, H., Preprocessing phase for Arabic word handwritten recognition. Russian Academy Of Sciences *Information Transmissions In Computer Networks*, 2006. **6**(1):pp. 11-19.
16. Al-Shatnawi, A. and Omar, K. , Detecting Arabic handwritten word baseline using Voronoi Diagram. In *Proceedings of the International Conference On Electrical Engineering And Informatics (ICEEI)*, 2009. pp. 18-22.
17. Al-Shatnawi, A. and Omar, K. , A comparative study between methods of Arabic baseline detection. In *Proceedings of the International Conference On Electrical Engineering And Informatics (ICEEI)*. 2009. pp. 73-77.
18. Al-Yousefi, H. and Upda, S. S., Recognition of Arabic characters, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1992. **14**(8): pp. 853-857.
19. Alamri, H., Recognition of off-line Arabic handwritten dates and numeral strings. 2009, Masters' Thesis, Department of Computer Science and Software Engineering, Concordia University.
20. Alamri, H., Sadri, J., Suen, C. Y. and Nobile, N. , A novel comprehensive database for Arabic off-line handwriting recognition. In *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008. pp. 664-669.
21. Alginahi, Y., A survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition (IJ DAR)*, 2013. **16**(2): pp. 105-126.
22. AlKhateeb, J. H., Ren, J., Ipson, S. S. and Jiang, J., Knowledge-Based Baseline Detection and Optimal Thresholding for Words Segmentation in Efficient Pre-Processing of Handwritten Arabic Text. In *Proceedings of the 5th International Conference on Information Technology: New Generations (ITNG)*, 2008. pp. 1158-1159.
23. AlKhateeb, J. H., Ren, J., Ipson, S. S. and Jiang, J. , Component-based segmentation of words from handwritten Arabic text. *International Journal of Computer Systems Science and Engineering*, 2009. **5**(1): pp. 309-313.
24. Amin, A., Al-Sadoun, H. and Fischer, S., Hand-printed Arabic character recognition system using an artificial network. *Pattern Recognition*, 1996. **29**(4): pp. 663-675.

25. Amin, A., Recognition of hand-printed characters based on structural description and inductive logic programming. *Pattern Recognition Letters*, 2003. **24**(16): pp. 3187-3196.
26. Amin, A., Recognition of printed Arabic text based on global features and decision tree learning techniques. *Pattern Recognition*, 2000. **33**(8): pp. 1309-1323.
27. Amin, A., Recognition of Printed Arabic Text via Machine Learning, *In Proceedings of the International Conference on Advances in Pattern Recognition*, S. Singh, Editor. 1999, Springer London. pp. 317-326.
28. Andrew, W. S. and J. R. Anthony, An Off-Line Cursive Handwriting Recognition System. *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, 1998. **20**(3): pp. 309-321.
29. Arica, N. and Yarman-Vural, F. T., Optical character recognition for cursive handwriting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002. **24**(6): pp. 801-813.
30. Belaid, A. and Choisy, C., Human Reading Based Strategies for Off-Line Arabic Word Recognition, *In Summit on Arabic and Chinese Handwriting Recognition*, D. Doermann and S. Jaeger, Editors. 2008, Springer Berlin Heidelberg. pp. 36-56.
31. Boukerma, H. and Farah, N., A Novel Arabic Baseline Estimation Algorithm Based on Sub-Words Treatment. *In Proceedings of the International Conference On Frontiers In Handwriting Recognition (ICFHR)*, 2010. pp. 335-338.
32. Boukerma, H. and Farah, N., Preprocessing Algorithms for Arabic Handwriting Recognition Systems. *In Proceedings of the International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, 2012. pp. 318-323.
33. Bozinovic, R. M. and Srihari, S. N., Off-line cursive script word recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1989. **11**(1): pp. 68-83.
34. Brown, M. K. and Ganapathy, S., Preprocessing techniques for cursive script word recognition. *Pattern Recognition*, 1983. **16**(5): pp. 447-458.
35. Bunke, H., Roth, M. and Schukat-Talamazzini, E.G., Off-line cursive handwriting recognition using hidden Markov models. *Pattern Recognition*, 1995. **28**(9): pp. 1399-1413.
36. Burr, D. J., A normalizing transform for cursive script recognition. *In Proceedings of 6th International Conference on Pattern Recognition (ICPR)*, 1982. pp. 1027-1030.
37. Burrow, P., Arabic handwriting recognition. Report of Master of Science, School of Informatics, University of Edinburgh, 2004.

38. Bushofa, B. M. F. and Spann, M. , Segmentation and recognition of Arabic characters by structural classification. *Image and Vision Computing*, 1997. **15**(3): pp. 167-179.
39. Caesar, T., Gloger, J. M. and Mandler, E., Preprocessing and feature extraction for a handwriting recognition system. *In Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR)*, 1993. pp. 408-411.
40. Caesar, T., Gloger, J. M. and Mandler, E., Estimating the baseline for written material, *In Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR)*. 1995, IEEE Computer Society. pp. 382-385.
41. Chan, J., Ziftci, C. and Forsyth, D., Searching Off-line Arabic Documents. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006. pp. 1455-1462.
42. Chang C. and Lin, C., LIBSVM: A Library for Support Vector Machines, 2001, available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
43. Cheriet, M., Visual Recognition of Arabic Handwriting: Challenges and New Directions, *In Arabic and Chinese Handwriting Recognition*, D. Doermann and S. Jaeger, Editors. 2008, Springer Berlin Heidelberg. pp. 1-21.
44. Cheriet, M. and Moghaddam, R., A Robust Word Spotting System for Historical Arabic Manuscripts, *In Guide to OCR for Arabic Scripts*, V. Margner and H. El Abed, Editors. 2012, Springer London. pp. 453-484.
45. Dalal, S. and Malik, L. , A Survey of Methods and Strategies for Feature Extraction in Handwritten Script Identification. *In Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology (ICETET)*, 2008. pp. 1164-1169.
46. Dehghan, M. and Faez, K., Farsi handwritten character recognition with moment invariants. *In Proceedings of the 13th International Conference on Digital Signal Processing Proceedings (DSP)* 1997. pp. 507-508.
47. Dehghan, M., Faez, K., Ahmadi, M. and Shridhar, M., Unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov models. *Pattern Recognition Letters*, 2001. **22**(2): pp. 209-214.
48. Dehghan, M., Faez, K., Ahmadi, M. and Shridhar, M., Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM. *Pattern Recognition*, 2001. **34**(5): pp. 1057-1065.
49. Ding, X. and Liu, H. , Segmentation-Driven Offline Handwritten Chinese and Arabic Script Recognition, *In Arabic and Chinese Handwriting Recognition*, D. Doermann and S. Jaeger, Editors. 2008, Springer Berlin Heidelberg. pp. 196-217.

50. El-Hajj, R., Mokbel, C. and Likforman-Sulem, L., Recognition of Arabic handwritten words using contextual character models. *In Proceedings of the SPIE 6815 Document Recognition and Retrieval XV*. 2008. pp. 681503_1-681503_9.
51. El-Hajj, R., Likforman-Sulem, L. and Mokbel, C. , Arabic handwriting recognition using baseline dependant features and hidden Markov modeling. *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*. 2005. pp. 893 - 897.
52. El-Sherif, E.A. and Abdelazeem, S. , A Two-Stage System for Arabic Handwritten Digit Recognition Tested on a New Large Database. *In Artificial Intelligence and Pattern Recognition*. 2007.pp. 237-242.
53. Elanwar, R. I., Rashwan, M. and Mashali, S. , Arabic online word extraction from handwritten text using SVM-RBF classifiers decision fusion, *In Proceedings of the 4th WSEAS international conference on Nanotechnology*. 2012. pp. 68-73.
54. Eraqi, H. M. and Abdelazeem, S. , HMM-based Offline Arabic Handwriting Recognition: Using New Feature Extraction and Lexicon Ranking Techniques. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 2012. pp. 555-559.
55. Erlandson, E. J., Trenkle, J. M. and Vogt, R. C., Word-level recognition of multifont Arabic text using a feature vector matching approach. *In Proceedings of the SPIE 2660 Document Recognition and Retrieval III*, 1996. pp. 63-70.
56. Espana-Boquera, S., Castro-Bleda, M. J., Gorbe-Moya, J. and Zamora-Martinez, F., Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011. **33**(4): pp. 767-779.
57. Faisal, F., Venu, G. and Michael, P. , Pre-processing Methods for Handwritten Arabic Documents, *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*. 2005, IEEE Computer Society. pp. 267-271.
58. Feldbach, M. and Tonnie, K. D., Word segmentation of handwritten dates in historical documents by combining semantic a-priori-knowledge with local features. *In Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR)*. 2003. pp. 333-337.
59. Garnes, S. J. A., Morita, M. E., Facon, J., Sampaio, R. J. B., Bortolozzi, F. and Saborin, R., Correcting handwriting baseline skew using direct methods on the weighted least squares. *XXXI Simpsio Brasileiro de Pesquisa Operacional*, Juiz de Fora, 1999. pp. 184-196.
60. Gatos, B. and Ntirogiannis, K. , Restoration of arbitrarily warped document images based on text line and word detection, *In Proceedings of the 4th conference on IASTED*

- International Conference: Signal Processing, Pattern Recognition and Applications*. 2007, ACTA Press: Innsbruck, Austria. pp. 203-208.
61. Gatos, B., Stamatopoulos, N. and Louloudis, G., ICFHR 2010 Handwriting Segmentation Contest. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 2010. pp. 737-742.
 62. Gatos, B., Stamatopoulos, N. and Louloudis, G., ICDAR2009 handwriting segmentation contest. *International Journal on Document Analysis and Recognition (IJ DAR)*, 2011. **14**(1): pp. 25-33.
 63. Gatos, B., Antonacopoulos, A. and Stamatopoulos, N., ICDAR2007 Handwriting Segmentation Contest. *In Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007. pp. 1284-1288.
 64. Gheith, A. A., Khaled, S. Y. and Mohammed, Z. K., Handwritten Arabic character recognition using multiple classifiers based on letter form, *In Proceedings of the 5th IASTED International Conference on Signal Processing, Pattern Recognition and Applications*. 2008, ACTA Press: Innsbruck, Austria. pp. 128-133.
 65. Gorbe-Moya, J., Boquera, S. E., Zamora-Martínez, F. and Bleda, M. J. C., Handwritten Text Normalization by using Local Extrema Classification. *In Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems (PRIS)*, 2008. **8**: pp. 164-172.
 66. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H. and Schmidhuber, J., A Novel Connectionist System for Unconstrained Handwriting Recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2009. **31**(5): pp. 855-868.
 67. Gyeonghwan, K., Govindaraju, V. and Srihari, S. N., A segmentation and recognition strategy for handwritten phrases. *In Proceedings of the 13th International Conference on Pattern Recognition*, 1996. pp. 510-514.
 68. Haji, M. M., Sahoo, K. A., Bui, T. D., Suen, C. Y. and Ponson, D., Statistical Hypothesis Testing for Handwritten Word Segmentation Algorithms. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012. pp. 114-119.
 69. Hasselberg, A., Zimmermann, R., Kraetzer, C., Scheidat, T., Vielhauer, C. and Kümmel, K., Security of Features Describing the Visual Appearance of Handwriting Samples Using the Bio-hash Algorithm of Vielhauer against an Evolutionary Algorithm Attack, *In Communications and Multimedia Security*, B. De Decker, et al., Editors. 2013, Springer Berlin Heidelberg. pp. 85-94.
 70. Hennig, A. and Sherkat, N. , Exploiting zoning based on approximating splines in cursive script recognition. *Pattern Recognition*, 2002. **35**(2): pp. 445-454.

71. Hennig, A., Sherkat, N. and Whitrow, R. J., Zone estimation for multiple lines of handwriting using approximating spline functions. *In Proceedings of the 5th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*. 1996: Citeseer. pp. 325-328
72. Houcine, B., Monji, K. and Adel, M. A., New Algorithm of Straight or Curved Baseline Detection for Short Arabic Handwritten Writing, *In Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR)*. 2009, IEEE Computer Society. pp. 778-782.
73. Huang, C. and Srihari, S. N., Word segmentation of off-line handwritten documents. *In Proceedings of the Document Recognition and Retrieval (DRR) XV,IST/SPIE Annual Symposium*, 2008.
74. Jamal, A. T., Nobile, N. and Suen, C. Y., End-Shape Recognition for Arabic Handwritten Text Segmentation, *In Artificial Neural Networks in Pattern Recognition*, N. El Gayar, F. Schwenker and C. Y. Suen, Editors. 2014, Springer International Publishing. pp. 228-239.
75. Jamal, A. T., Nobile, N. and Suen, C. Y. , Learning-based Baseline Estimation, *In Proceedings of the 11th International Conference on Pattern Recognition and Image Analysis (PRIA)*. 2013. Samara, Russia. pp. 583-586.
76. Jamal, A. T. and Suen, C. Y. , Removal of Secondary Components of Arabic Handwritten Words Using Morphological Reconstruction, *In Proceedings of the International Conference on Information Technology (ICIT)*. 2014. Dubai, UAE. pp. 10.
77. Karacs, K., Pr3sz3ky, G. and Roska, T. , Cellular wave computer algorithms with spatial semantic embedding for handwritten text recognition. *International Journal of Circuit Theory and Applications*, 2009. **37**(10): pp. 1019-1050.
78. Kchaou, M. G., Kanoun, S. and Ogier, J. M., Segmentation and Word Spotting Methods for Printed and Handwritten Arabic Texts: A Comparative Study. *In Proceedings of the International Conference on Frontiers Handwriting Recognition (ICFHR)*, 2012. pp. 274-279.
79. Kharma, N., Ahmed, M. and Ward, R., A new comprehensive database of handwritten Arabic words, numbers and signatures used for OCR testing. *In Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, 1999. pp. 766-768.
80. Khayyat, M., Lam, L. and Suen, C. Y. , Learning-based word spotting system for Arabic handwritten documents. *Pattern Recognition*, 2014. **47**(3): pp. 1021-1030.
81. Khayyat, M., Lam, L., Suen, C. Y., Yin, F. and Liu, C. L. , Arabic Handwritten Text Line Extraction by Applying an Adaptive Mask to Morphological Dilation. *In Proceedings the 10th IAPR International Workshop on Document Analysis Systems (DAS)*, 2012. pp. 100-104

82. Khayyat, M., Lam, L. and Suen, C. Y. , Verification of Hierarchical Classifier Results for Handwritten Arabic Word Spotting. *In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013. pp. 572-576.
83. Khayyat, M., Lam, L. and Suen, C. Y. , Arabic handwritten word spotting using language models. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012. pp. 43-48.
84. Khedher, M. Z., Abandah, G. A. and Al-Khawaldeh, A. M., Optimizing Feature Selection for Recognizing Handwritten Arabic Characters. *In Proceedings of world academy of science, engineering and technology*, 2005: Citeseer. pp. 81-84.
85. Khedher, M. Z. and Abandah, G. , Arabic character recognition using approximate stroke sequence. *In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, 2002. pp. 28-34.
86. Khorsheed, M.S. and Clocksin, W.F. , Multi-font Arabic word recognition using spectral features. *In Proceedings of the 15th International Conference on Pattern Recognition (ICPR)*, 2000. pp. 543-546.
87. Kim, G. and Govindaraju, V. , Handwritten phrase recognition as applied to street name images. *Pattern Recognition*, 1998. **31**(1): pp. 41-51.
88. Kim, G., Govindaraju, V. and Srihari, S. N., An architecture for handwritten text recognition systems. *International Journal on Document Analysis and Recognition*, 1999. **2**(1): pp. 37-44.
89. Lawgali, A., Bouridane, A. , Angelova, M. and Ghassemlooy, Z. , Handwritten Arabic character recognition: Which feature extraction method? *International Journal of Advanced Science and Technology*, 2011. **34**: pp. 1-8.
90. Lemaitre, A. l., Camillerapp, J. and Couasnon, B., A perceptive method for handwritten text segmentation. *In Proceedings of SPIE 9021 Document Recognition and Retrieval XVIII*. 2011. pp. 78740C_1- 78740C_9.
91. Leydier, Y., Lebourgeois, F. and Emptoz, H., Text search for medieval manuscript images. *Pattern Recognition*, 2007. **40**(12): pp. 3552-3567.
92. Likforman-Sulem, L., Mohammad, R. A. H., Mokbel, C., Menasri, F., Bianne-Bernard, A. L. and Kermorvant, C., Features for HMM-Based Arabic Handwritten Word Recognition Systems, *in Guide to OCR for Arabic Scripts*, V. Margner and H. El Abed, Editors. 2012, Springer London. pp. 123-143.
93. Liwicki, M. and Bunke, H., Combining diverse on-line and off-line systems for handwritten text line recognition. *Pattern Recognition*, 2009. **42**(12): pp. 3254-3263.
94. Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C., Text line and word segmentation of handwritten documents. *Pattern Recognition*, 2009. **42**(12): pp. 3169-3183.

95. Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C., Line and word segmentation of handwritten documents. *In Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 2008. pp.247–252.
96. Luthy, F., Varga, T. and Bunke, H. , Using Hidden Markov Models as a Tool for Handwritten Text Line Segmentation. *In Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007. pp. 8-12.
97. Maddouri, S. S., Samoud, F. B., Bouriel, K., Ellouze, N. and El Abed, H., Baseline extraction: Comparison of six methods on ifn/enit database. *In Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008. pp. 571–576.
98. Mahadevan, U. and Nagabushnam, R. C., Gap metrics for word separation in handwritten lines. *In Proceedings of the 3rd International Conference on in Document Analysis and Recognition (ICDAR)*, 1995. pp. 124-127.
99. Mahmoud, S. A., Ahmad, I., Al-Khatib, W. G., Alshayeb, M., Parvez, M. T., Märgner, V. and Fink, G. A., KHATT: An open Arabic offline handwritten text database. *Pattern Recognition*, 2014. **47**(3): pp. 1096-1112.
100. Manmatha, R. and Rothfeder, J. L. , A scale space approach for automatically segmenting words from historical handwritten documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2005. **27**(8): pp. 1212-1225.
101. Manmatha, R. and Srimal, N., Scale Space Technique for Word Segmentation in Handwritten Documents, *in Scale-Space Theories in Computer Vision*, M. Nielsen, et al., Editors. 1999, Springer Berlin Heidelberg. pp. 22-33.
102. Manmatha, R., Chengfeng, H. and Riseman, E.M., Word spotting: a new approach to indexing handwriting. *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996. pp. 631-637.
103. Marti, U.V. and Bunke, H., Text line segmentation and word recognition in a system for general writer independent handwriting recognition. *In Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR)*, 2001. pp. 159-163.
104. Menasri, F., Vincent, N., Augustin, E. and Cheriet, M., Shape-Based Alphabet for Off-line Arabic Handwriting Recognition. *In Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007. pp. 969 - 973.
105. Menasri, F., Vincent, N., Augustin, E. and Cheriet, M., Un systeme de reconnaissance de mots arabes manuscrits hors-ligne sans signes diacritiques. *In Colloque International Francophone sur l'Ecrit et le Document*. 2008: Groupe de Recherche en Communication Ecrite. pp. 121-126.

106. Moghaddam, R. F. and Cheriet, M., Application of Multi-Level Classifiers and Clustering for Automatic Word Spotting in Historical Document Images. *In Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR)*, 2009. pp. 511 - 515.
107. Motawa, D., Amin, A. and Sabourin, R. , Segmentation of Arabic cursive script. *In Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR)*, 1997. pp. 625-628.
108. Mowlaei, A., Faez, K. and Haghghat, A. T., Feature extraction with wavelet transform for recognition of isolated handwritten Farsi/Arabic characters and numerals. *In Proceedings of the 14th International Conference on Digital Signal Processing (DSP)*, 2002. pp. 923-926.
109. Mowlaei, A. and Faez, K. , Recognition of isolated handwritten Persian/Arabic characters and numerals using support vector machines. *In Proceedings of the 13th IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, 2003. pp. 547-554.
110. Mozaffari, S., Faez, K. and Rashidy Kanan, H., Recognition of isolated handwritten Farsi/Arabic alphanumeric using fractal codes. *In Proceedings of the 6th IEEE Southwest Symposium on Image Analysis and Interpretation*, 2004. pp. 104-108.
111. Mozaffari, S., Faez, K. and Rashidy Kanan, H., Feature comparison between fractal codes and wavelet transform in handwritten alphanumeric recognition using SVM classifier. *In Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, 2004. pp. 331 - 334.
112. Nadi, F., Sadri, J. and Foroozandeh, A., A novel method for slant correction of Persian handwritten digits and words. *In Proceedings of the First Iranian Conference on Pattern Recognition and Image Analysis (PRIA)*, 2013. 2013. pp. 295-298.
113. Nagabhushan, P. and Alaei, A. , Tracing and straightening the baseline in handwritten persian/arabic text-line: A new approach based on painting-technique. *International Journal on Computer Science and Engineering*, 2010. 2(4): pp. 907-916.
114. Natarajan, P., Saleem, S., Prasad, R., MacRostie, E. and Subramanian, K., Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach, *in Arabic and Chinese Handwriting Recognition*, D. Doermann and S. Jaeger, Editors. 2008, Springer Berlin Heidelberg. pp. 231-250.
115. Naz, S., Razzak, M. I., Hayat, K., Anwar, M. W. and Khan, S. Z., Challenges in Baseline Detection of Arabic Script Based Languages, *in Intelligent Systems for Science and Information*, L. Chen, S. Kapoor and R. Bhatia, Editors. 2014, Springer International Publishing. pp. 181-196.

116. Naz, S., Hayat, K., Anwar, M. W., Akbar, H. and Razzak, M. I., Challenges in baseline detection of cursive script languages. *In Proceedings of the Science and Information Conference (SAI)*, 2013. 2013. pp. 551-556.
117. Olivier, G., Miled, H., Romeo, K. and Lecourtier, Y., Segmentation and coding of Arabic handwritten words. *In Proceedings of the 13th International Conference on Pattern Recognition (ICPR)*, 1996. pp. 264-268.
118. Otsu, N., A threshold selection method from gray-level histogram, *In: IEEE Trans. System Man Cybernetics*. 1979. 9(1): pp. 1569-1576.
119. Ouwayed, N. and Belaïd, A., Multi-oriented Text Line Extraction from Handwritten Arabic Documents. *In Proceedings of the 8th IAPR International Workshop on Document Analysis Systems (DAS)*, 2008. pp. 339 - 346.
120. Öztop, E., Mülayim, A. Y., Atalay, V. and Yarman-Vural, F., Repulsive attractive network for baseline extraction on document images. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997. pp. 3181 - 3184.
121. Papavassiliou, V., Stafylakis, T., Katsouros, V. and Carayannis, G., Handwritten document image segmentation into text lines and words. *Pattern Recognition*, 2010. **43**(1): pp. 369-377.
122. Park, J. and Govindaraju, V. , Use of adaptive segmentation in handwritten phrase recognition. *Pattern Recognition*, 2002. **35**(1): pp. 245-252.
123. Parvez, M. T. and Mahmoud, S. A. , Arabic handwritten alphanumeric character recognition using fuzzy attributed turning functions. *In Proceedings of the First International Workshop on Frontiers in Arabic Handwriting Recognition*, 2011. pp. 9-14.
124. Patrice, Y. S., Dave, S. and Maneesh, A. , Ink Normalization and Beautification, *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*. 2005, IEEE Computer Society. pp. 1182 - 1187.
125. Pechwitz, M. and Margner, V. , Baseline estimation for Arabic handwritten words. *In Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, 2002. pp. 479 - 484.
126. Pechwitz, M., El Abed, H. and Margner, V., Handwritten Arabic Word Recognition Using the IFN/ENIT-database, *in Guide to OCR for Arabic Scripts*, V. Margner and H. El Abed, Editors. 2012, Springer London. pp. 169-213.
127. Pechwitz, M., Maddouri, S. S., Märgner, V., Ellouze, N. and Amiri, H., IFN/ENIT-database of handwritten Arabic words. *In Proceedings of CIFED*. 2002: Citeseer. pp. 129-136.

128. Rodríguez-Serrano, J. A. and Perronnin, F., A Model-Based Sequence Similarity with Application to Handwritten Word Spotting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2012. **34**(11): pp. 2108-2120.
129. Sadri, J., Suen, C. Y. and Bui, T. D. , Automatic segmentation of unconstrained handwritten numeral strings. *In Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2004. pp. 317–322.
130. Sagheer, M. W., Novel word recognition and word spotting systems for offline Urdu handwriting. Master Thesis, 2010, Department of Computer Science and Software Engineering ,Concordia University.
131. Sahlol, A. T., Suen, C. Y., Sallam, A.A., Elbasyoni, M.R.: A proposed OCR Algorithm for cursive Handwritten Arabic Character Recognition. *Journal of Pattern Recognition and Intelligent Systems (PRIS)*, 2014. pp. 90–104.
132. Sahlol, A. T., Suen, C. Y., Elbasyoni, M. R. and Sallam, A. A., Investigating of Preprocessing Techniques and Novel Features in Recognition of Handwritten Arabic Characters, *in Artificial Neural Networks in Pattern Recognition*, N. El Gayar, F. Schwenker and C. Y. Suen, Editors. 2014, Springer International Publishing. p. 264-276.
133. Sánchez, A., Mello, C. A., Suárez, P. D. and Lopes, A., Automatic line and word segmentation applied to densely line-skewed historical handwritten document images. *Integrated Computer Aided Engineering Journal Impact Factor & Information*, 2011. **18**(2): pp. 125-142.
134. Sari, T. and Kefali, A., A search engine for Arabic documents. *in Colloque International Francophone sur l'Ecrite et le Document*. 2008: Groupe de Recherche en Communication Ecrite. pp. 97-102.
135. Sarkar, A., Biswas, A., Bhowmick, P. and Bhattacharya, B. B., Word Segmentation and Baseline Detection in Handwritten Documents Using Isothetic Covers. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2010. pp. 445 - 450.
136. Saykol, E., Sinop, A. K., Gudukbay, U., Ulusoy, O. and Çetin, A. E., Content-based retrieval of historical Ottoman documents stored as textual images. *Image Processing, IEEE Transactions on*, 2004. **13**(3): pp. 314-325.
137. Seni, G. and Cohen, E., External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 1994. **27**(1): pp. 41-52.
138. Shahab, S. A., Al-Khatib, W. G. and Mahmoud, S. A., Computer Aided Indexing of Historical Manuscripts. *In Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation*, 2006. pp. 287 - 295.

139. Simistira, F., Papavassiliou, V., Stafylakis, T. and Katsouros, V., Enhancing Handwritten Word Segmentation by Employing Local Spatial Features. *In Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR)*, 2011. pp. 1314 - 1318.
140. Slimane, F., Hennebert, S. K. J. and Alimi, R. I. A. M. , A New Baseline Estimation Method Applied to Arabic Word Recognition. *In Proceedings of the 10th IAPR International Workshop on Document Analysis Systems (DAS)*. 2012.
141. Sridha, M., Mandalapu, D. and Patel, M., Active-dtw: a generative classifier that combines elastic matching with active shape modeling for online handwritten character recognition. *In Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*. 2006: Suvisoft. pp. 193–196.
142. Srihari, S., Srinivasan, H., Babu, P. and Bhole, C., Spotting words in handwritten Arabic documents. *In Proceedings of SPIE 6067 Document Recognition and Retrieval XIII* 2006. pp. 606702-1-606702-12.
143. Srihari, S. and Ball, G. , Language Independent Word Spotting in Scanned Documents, *in Digital Libraries: Universal and Ubiquitous Access to Information*, G. Buchanan, M. Masoodian and S. Cunningham, Editors. 2008, Springer Berlin Heidelberg. pp. 134-143.
144. Stafylakis, T., Papavassiliou, V., Katsouros, V. and Carayannis, G., Robust text-line and word segmentation for handwritten documents images. *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008. pp. 3393–3396.
145. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U. and Alaei, A., ICDAR 2013 Handwriting Segmentation Contest. *In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013. pp. 1402-14-6.
146. Strassel, S., Linguistic resources for Arabic handwriting recognition. *In Proceedings of the 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt. 2009: Citeseer. pp. 37-41.
147. Sun, Y., Butler, T. S., Shafarenko, A., Adams, R., Loomes, M. and Davey, N. Identifying word boundaries in handwritten text. *In Proceedings of the International Conference on Machine Learning and Applications*, 2004. pp. 5-9.
148. Sun, Y., Butler, T. S., Shafarenko, A., Adams, R., Loomes, M. and Davey, N., Word segmentation of handwritten text using supervised classification techniques. *Applied Soft Computing*, 2007. 7(1): pp. 71-88.
149. Varga, T. and Bunke, H., Tree structure for word extraction from handwritten text lines. *In Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, 2005. pp. 352 - 356.

150. Vincent, L., Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *Image Processing, IEEE Transactions on*, 1993. **2**(2): pp. 176-201.
151. Vinciarelli, A. and Luetttin, J., A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 2001. **22**(9): pp. 1043-1050.
152. Wienecke, M., Fink, G. A. and Sagerer, G. , Towards automatic video-based whiteboard reading. *In Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR)*, 2003. pp. 188-200.
153. Wshah, S., Kumar, G. and Govindaraju, V., Script Independent Word Spotting in Offline Handwritten Documents Based on Hidden Markov Models. *In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012. pp. 14-19.
154. Zaghoul, R. I., AlRawashdeh, E. F. and Bader, D. M. K., Multilevel Classifier in Recognition of Handwritten Arabic Characters. *Journal of Computer Science*, 2011. **7**(4): pp. 512.
155. Zamora-Martínez, F., Castro-Bleda, M. J., España-Boquera, S. and Gorbe-Moya, J., Unconstrained offline handwriting recognition using connectionist character N-grams. *The Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2010. pp. 1-7.
156. Ziaratban, M. and Faez, K. , A novel two-stage algorithm for baseline estimation and correction in Farsi and Arabic handwritten text line. *In Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*, 2008. pp. 1-5.
157. Ziaratban, M. and Faez, K. , Non-uniform slant estimation and correction for Farsi/Arabic handwritten words. *International Journal on Document Analysis and Recognition (IJDAR)*, 2009. **12**(4): pp. 249-267.
158. Ziaratban, M., Faez, K. and Allahveiradi, F., Novel Statistical Description for the Structure of Isolated Farsi/Arabic Handwritten Characters. *In Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008. pp. 332-337.
159. Zimmermann, M. and Bunke, H. , Automatic segmentation of the IAM off-line database for handwritten English text. *In Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*, 2002. pp. 35-39.
160. Dumais, S., Using SVMs for text categorization. *IEEE Intelligent Systems*, 1998. **13**(4): pp. 21-23.

161. Gatos, B., Pratikakis, I., Kesidis, A. L. and Perantonis, S. J., Efficient off-Line cursive handwriting word recognition. *In the Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2006. pp. 121-125.
162. Zhang, T. Y. and Suen C. Y., A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 1984. **27**(3): pp. 236-239.
163. Vaseghi, B. and Hashemi, S., Farsi handwritten word recognition using discrete HMM and self-organizing feature map. *In Proceedings of the International Computer Science and Information Technology*, 2012. pp. 123-129.