DEVELOPMENT OF PART-OF-ARABIC-WORD CORPUS FOR HANDWRITING TEXT RECOGNITION

BY

HASAN HADDAD HAMED AL-KAF

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

MAY 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **HASAN HADDAD HAMED AL-KAF** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN INFORMATION AND COMPUTER SCIENCE.**

Thesis Committee

Dr. Wasfi G. Al-Khatib (Adviser)

(Co-adviser)

Mahmoud

Dr. Sabri Mahmoud (Member)

R E Abdel Aal

Dr. Radwan E. Abdel-Aal (Member)

(Member)

Dr. Abdulaziz Alkhoraidly
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

29/12/15

Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

To my Parents, my wife and my daughters Noor & Mahani
for their endless love,support and encouragement

ACKNOWLEDGMENTS

In the name of Allah, most Kind, most Merciful. Thanks are to Allah who gave me strength, patience and ability to accomplish this research. I would like to thank my family who supported me during my study. I am particularly grateful to my wife who supported me during this work. I would like to thank King Fahd University for Petroleum and Minerals and Seiyun Community College for giving me the opportunity to complete my study in KFUPM university.

No words can express my thanks to my thesis advisor, Dr. Wasfi G. Al-Khatib, for his guidance, support and advice.

I would like also to express my sincere gratitude to Dr. Sabri Mahmoud for his support and advice. Thanks are due to my thesis committee member Dr. Radwan E. Abdel-Aal for his cooperation, valuable comments, and support.

I would like to, also, thank the authors of IFN/ENIT-database [1] for providing me with their database and inspiring me throught my work.

Thanks are also due to Mr.Osama AL-Dafer for his help in PAWs creation using the framework of the work. I would like to thank Mr. Karim who provided support for High Performance Computing(HPC) Of KFUPM Information Technology Center (ITC).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AHTR	Arabic Handwritten Text Recognition
PAWs	Part of Arabic Words
CPG	Circular Polar Grid
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
OCR	Optical Character Recognition

THESIS ABSTRACT

NAME:	Hasan Haddad Hamed Al-Kaf
TITLE OF STUDY:	Development of Part-of-Arabic-Word Corpus for Handwriting Text Recognition
MAJOR FIELD:	INFORMATION AND COMPUTER SCIENCE
DATE OF DEGREE:	May 2015

Arabic handwritten text recognition (AHTR) has become one of the interesting topics for researchers in recent years due to the increased need to digitize and store these documents in textual form. This thesis focuses on offline AHTR based on part of Arabic words (PAWs). A semi-automatic framework of PAWs extraction and ground-truthing was developed, that can be used as a basis for building PAW benchmark data. PAWs clustering and PAWs recognition were developed for handwritten text documents. Line images of KHATT-Database and IFN/ENIT-database were used in our experimentation. We analyzed the generated PAWs and studied the frequency and size of different PAWs for the sake of PAW recognition. Different machine-learning techniques were investigated to carry out PAW Identification, such as the K-nearest Neighbor with values 1, 3 and 5 for K and support vector machine (SVM). Moreover, a variety of Circular Polar Grid (CPG) that is generated from a set of four low-level features and Gabor features were used in order to choose the most successful ones that will result in the best Correctly Classified Instances. One thousand classes were used from KHATT-Database and 381 from IFN/ENIT-database. The best achieved result was 73.07% of correctly classified instances using only IFN/ENIT-database with a mix of Gabor and CPG features with SVM classifier.

ملخص الرسالة

الاسم الكامل : حسن حداد حامد الكاف

عنوان الرسالة : تطوير قاعدة بيانات للأجزاء العربية المتصلة المكتوبة بخط اليد للتعرف على النص

التخصص : علوم الحاسب الالي

تاريخ الدرجة العلمية : مايو 2015

يعتبر التعرف على الخط العربي المكتوب باليد أحد المواضيع الشيقة للبحث في الوقت الراهن, ويعزى ذلك إلى الحاجة لاستخلاص النصوص العربية من الصور الرقمية وتمثيلها نصيا ، نقوم في هذه الرسالة بالتركيز على استرجاع الأجزاء النصية العربية (Part of Arabic Words) - والتي تتكون منها الكلمات – من الصور المخزنة مسبقا ، فقمنا بتطوير نظام أو إطار عمل مصغر (Framework) شبه تلقائي وذلك لاستخراج الأجزاء العربية المتصلة وتخزينها كصورة مع ما يقابلها من النص ، حيث يمكن استخدام هذا النظام في توليد قاعدة بيانات مختصة بهذه الأجزاء بأقل وقت ممكن وكفاءة عالية. كما قمنا بتطوير نظام لتصنيف هذه الأجزاء العربية المتصلة والتعرف عليها.

ولتوليد قاعدة البيانات هذه ، استخدمت الصور المحتوية على الأسطر المفردة من النص لقاعدة البيانات "خط" (KHATT-Database) وقاعدة البيانات (IFN/ENIT). بالإضافة الى ذلك قمنا بتحليل الأجزاء العربية المكتوبة بخط اليد التي تم توليدها ، كما درسنا تكرار وحجم هذه الإجزاء من أجل التعرف عليها آليا فيما بعد.

نعتمد في هذه الرسالة على تطويع تقنيات التعلم الآلي (Machine-Learning) في التعرف على الأجزاء العربية المتصلة حيث قمنا بدراسة تقنية أقرب عنصر مجاور (K-nearest Neighbor) و تقنية القوة الداعمة (SVM), كما تمت تجربة ميزة الشبكة القطبية الدائرية (Circular Polar Grid) الناتجة من مجموعة مميزات حسابية بسيطة ، وميزة لوغاريثم غابور (Gabor) ، وتم استخدام 1000 تصنيف (Class) مختلف من البيانات المجمعة ، حيث تم استرداد ما يقرب من (73.07%) من إجمالي الأجزاء المختلفة الموجودة في الصور بواسطة تقنية القوة الداعمة مستخدمين لخليط من ميزات لوغاريثم غابور والشبكة القطبية الدائرية.

CHAPTER 1

INTRODUCTION

1.1 Background

Segmentation of Arabic Handwriting Text is a crucial step and one of the most challenging problems in automatic optical character recognition (OCR). It leads to errors in recognition because its cursiveness [3]. Handwriting text recognition can be carried out in an online or an offline manner [4]. One attempts is to use character segmentation of Arabic text. Compared to Arabic printed text, character recognition of Arabic handwritten text remains a challenging problem in pattern recognition [5]. Text recognition of handwritten documents is difficult to perform manually and it would need huge amounts of human resources to execute on hundreds of thousands of documents. Furthermore, there is a growing need for recognition systems for Arabic text. Due to the existence of a huge amount of historical documents and unreadable text images, the need for text recognition in manuscripts is higher than that in printed text. Similarly, recent handwritten text documents may also be difficult to read, the reason is either during text image scanning or during text writing, although they are easier than their historical counterparts.

Arabic script has distinctive features than other scripts. For example, Arabic script is cursive and is written horizontally from right to left. Words consist of connected components or sub-words that are often called Part of Arabic Words (PAWs) [6, 7]. Hence, word spotting in Arabic can be carried out at the word level or the PAW level. In fact, PAWs are used in the most common way of Arabic Handwritten Word Spotting [8] and a search engine for Arabic documents [6]. Hence, it will be interesting to investigate the use of PAW identification for Handwritten Arabic Text recognition. Line segmentation of text in images into words and sub words by assigning dots and diacritics to respective words or sub words need more improvement and sometimes are not applicable to Arabic Handwritten text. Segmentation of Arabic Handwritten Text differs from dataset to another depending on the clarity of writing text and image resolution. Some techniques gave high recognition rates on a specific data set, but lower recognition rates have been reported using the same techniques on other datasets [9].

Although, word spotting and handwriting recognition researches have been reported for Latin-based languages and Chinese [10] [11], few works were reported for Arabic. Some word spotting techniques are based on PAWs and language models using unigram, bigram, trigram and 4-gram with 69 words that have been segmented into their PAWs. The number of words with one, two, three and four

PAWs were 23, 23, 20 and 3, respectively. [8], while others used overlapped windows to segment line images [12]. The methodology of [13] for handwritten text recognition generates a set of rules that are applied in sequence to match a character tree. Figure 1.1 shows an example of Arabic handwritten text paragraph taken from KHATT-Database [2].

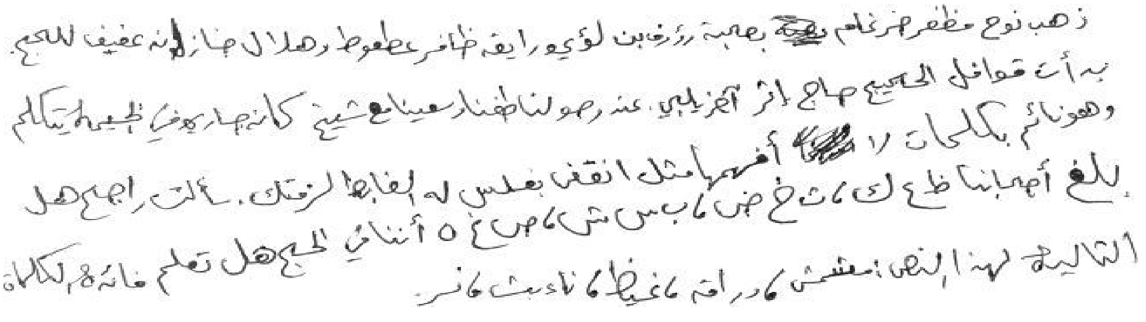


Figure 1.1: Sample Arabic handwritten text paragraph from KHATT-Database [2].

Arabic handwritten text recognition and word spotting have many challenges. Unclear boundaries between words or within a single word components that make word recognition a difficult task [8]. For example, Figure 1.2 shows small white spaces between the letter Saad ص and letter Haa ح and between the letter Yaa ي and letter Haa ح in the word Alsaheeha الصحيحة, that should not have existed within this single connected component. Another example in figure 1.3 that shows two words Wahowa Naaim وهو نائم with relatively long spacing between the sequence of components Naa نا and Em ئم within that single word, whereas the spacing between the two words Wahowa وهو and Naaim نائم is so small that one may think that this is a single word.

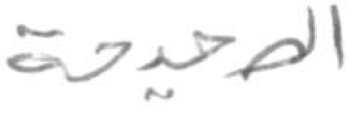


Figure 1.2: Small white spaces within single connected component.

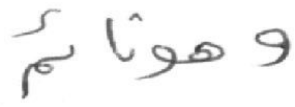


Figure 1.3: Unequal distance between two sequences of components or words.

Another major challenge of recognizing Arabic handwritten text is the existence of components of written words that are overlapped or scratched. For example, figure 1.4 has two overlapping or touching components, viz., Ayn ع is overlapping with Sheen ش and Saad ص is overlapping with Seen س. Also, the two words Afeef Llhaj عفيف للحج consisting of two connected components seem to consist of one connected component in figure 1.5, due to touching the last character Faa ف of the first word and the first character Laam لـ of the second word Llhajللحج.

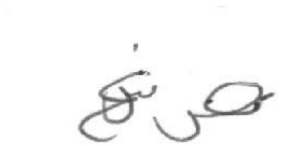


Figure 1.4: Components overlapped on each other.

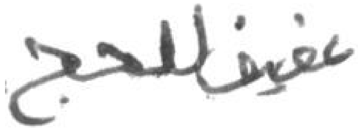


Figure 1.5: Continuous two sequenced components.

Last but not least, it is not uncommon to have certain components that are not related to the nearby word or PAW. Figures 1.6 and 1.7 show a small scratch that seems part of the word Adhabett الضابط and a stand alone scratched word that should be discarded, respectively.

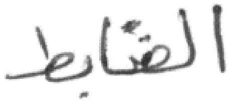


Figure 1.6: Extra components within a word.

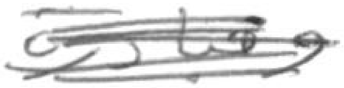


Figure 1.7: Scratch over the components.

1.2 Problem Statement

Part of Arabic Words (PAWs) identification is a major challenge in Arabic handwritten text recognition, which is a pattern recognition problem. Starting with PAW identification in carrying out handwritten text recognition is one choice for researchers to investigate, as it has the potential of avoiding character segmentation when recognizing handwritten Arabic text. The objectives of this thesis is to build a database of PAWs Images and their corresponding ground truth text in order to use them in handwritten text recognition. Given an input Arabic handwritten text in image form, we are segmenting the text into PAWs,clustering PAWs, extracting image features from each PAW, and finally recognizing them using suitable classifiers. The achievement of this goal is further explained in Section 1.3 of thesis Contributions. Figure 1.8 shows an example of Arabic Handwritten text and Table 1.1 shows its corresponding PAW-segmented text.

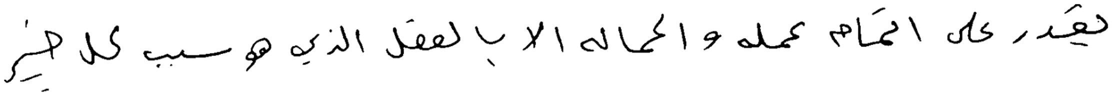


Figure 1.8: Arabic Handwritten Text Example.

Table 1.1: PAWs and their corresponding text

PAW Image											
PAW Text	له	كما	ا	و	عمله	م	تما	إ	على	ر	يقد
PAW Image											
PAW Text	خير	كل	سبب	هو	ي	لذ	ا	لعقل	با	لا	ا

Most research work on offline handwritten text recognition attempts the segmentation of text into words and carries out the recognition at the word level. In this thesis we study the feasibility of carrying out offline Arabic handwritten text recognition at the PAWs level. Our motivation for using PAWs is to avoid pre-segmentation. In order to achieve our goals, we need to estimate and/or find the number of different PAWs that occur in natural Arabic handwritten text. In order to achieve this, we utilize KHATT, a database of handwritten text developed at KFUPM [2]. Furthermore,we utilize IFN/ENIT-Database [1] of handwritten Arabic words which will be presented in the chapter 4. Second, we need to investigate various features and classifiers to carry out the recognition process.

1.3 Contributions Of The Thesis

In this thesis, we investigate the use of PAWs in Arabic handwriting text recognition. The following are the main processes to achieve this goal:

- Build a framework that facilitates PAWs creation from line images and build PAWs Database.
- Develop a PAW clustering system since different PAWs that are similar in shapes but only different in dots and /or diacritics need to be grouped together.
- Develop a PAW recognition system.

- Conduct Experimentation using developed PAWs of KHATT and IFN/ENIT databases.

In this thesis, we built A semi-automatic framework of PAWs extraction and ground-truthing and investigate the viability of using PAW in recent handwritten documents in order to automatically transcribe them at the line level. The work is organized as follows:

Chapter 1 introduced background of PAW Identification for Handwriting Text Recognition with problem definition and contributions. Chapter 2 presents a literature survey of word spotting problem and handwriting text recognition for handwritten documents. Chapter 3 describes our framework for PAW generation and recognition with adapted methodology to achieve the objectives of the work. Chapter 4 explains development of PAW corpus for KHATT and IFN/ENIT databases to be used in this work and chapter 5 describes the results of different uses combinations of databases, features and classifiers. Chapter 6 includes the conclusions, limitation and suggestions for future work.

CHAPTER 2

LITERATURE SURVEY

Handwriting text recognition received a lot of attention from researches in the past twenty years. Some of researches have been carried out for Latin-based languages, such as writer identification [14] [15] [16]or word spotting for English [17], and eastern languages such as Chinese [11] [18] [19] [20]. Lesser work has been reported for Arabic in last ten years and it has increased dramatically in the past few years [9]. Arabic writing, in general, is very different from most other languages. Arabic words are written in cursive where most of them may consist of one, two, three or more connected components. Each component is called part of Arabic word (PAW). In this literature survey, we present sample work carried out in word spotting, followed by a survey of research in handwriting text recognition in Arabic and Persian languages .

2.1 Word Spotting

Word spotting is a specialized part of handwritten text recognition that deals with the identification of keywords in handwritten text. Different approaches for word spotting of Arabic and other scripts have been reported [8] [12] [21].

Rath et al. proposed a method for word matching on George Washingtons manuscripts database [22]. The used features such as length, area, aspect ratio, number of ascenders & descenders, and the number of black to white transitions have been extracted. They analyzed four similarity measures, viz., sum of squared differences, Euclidean Distance Mapping, XOR algorithm and Scott & Longuet Higgins (SLH) algorithm. The SLH algorithm requires more processing time than other measures and is less robust in terms of noise and variation in size.

The research work of [21] proposed a method using Dynamic Time Wrapping (DTW) classifier with three different types of features to carry out word spotting for English handwritten text. The three features are: projection profile, word profile and background/ink transitions. They have shown that DTW is much faster than other classifiers like shape context matching. The proposed method matched and grouped the words into clusters where all instances of the same word are contained in the cluster. George Washingtons database of manuscript documents were used with 65% of them having good quality. The recognition rate was 72%.

In [23], the researchers used euclidean distance and histogram intersection values as similarity measures, with a database of historical documents from the Ottoman Empire archives developed by [24], which were written using Arabic script. They extracted the angular span, distance span and Wavelet domain features. The average precision of their results was 86%. It is worth noting that they applied those features on gray-level images, without using any binarization. In addition, their system can be applied to other scripts. Also, they reported that the classifiers they used have limited robustness to noise and variations in size.

In the work of [6], a search engine for textual dominant documents whether handwritten or printed has been proposed. They extracted the connected components of text, and extracted features related to loops, ascenders, descenders and diacritics, producing a symbolic representation for each connected component. They applied an approximate string matching technique based on Levenshtein distance and 123 historical document pages. For a set of queries they carried out, they reported 77.78% precision and 56.62% recall, although they did not provide the precision-recall graphs.

The work of [25] proposed a technique to segment and classify words of historical Arabic manuscript. They used structural and statistical features with Neural Networks to classify words. The training dataset consisted of 200 words that were selected manually containing 5 different writing styles where testing dataset consisted of 200 words that written by 5 different writing style, and a recognition rate of 89.3% was reported.

The thesis report of [7] proposed employing content based image retrieval (CBIR) techniques and latent semantic indexing (LSI) approach to index historical manuscripts. In addition to different features used, the author proposed using a feature called Circular Polar Grid (CPG) that is generated from a set of four low-level features, viz., concentric, rectangular, angular and concentric polar grid features. The query matching mechanisms are applied to retrieve candidate words and setting a threshold for similarity matching using LSI that is less dependent on the underlying features. The system tested on two historical Arabic manuscripts. 34 pages with 5500 words of first manuscript are used and the second manuscript was used for comparison purposes.

The results showed that the system was able to retrieve relevant words. The circular polar grid features have shown superiority over other feature sets. The reported recognition rate of his work was 78.7%.

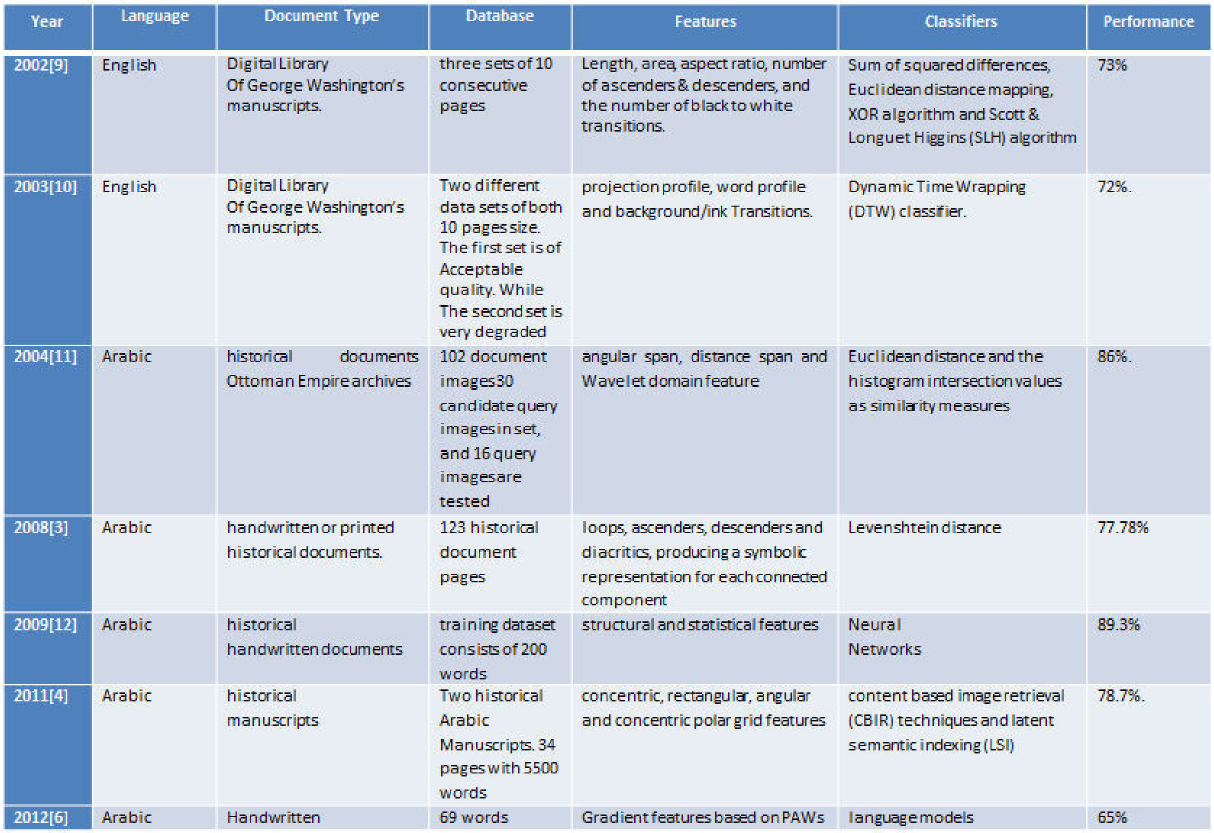
The work of [12] suggested that line-based keyword spotting is better than template or word-based keyword spotting. In the line-based approach, the word or character segmentation occurs during the spotting process, unlike the previous two approaches, in which segmentation precedes the spotting. They employ Hidden Markov Models using character and lexicon based background models. They extracted the features from each segmented line using 85%-overlapped windows of width 20 pixels. Each window was further segmented into bins in order to extract gradient and intensity features. For their experimental results, they used three public dataset, viz., the IAM dataset for English [26], the AMA dataset for

Arabic [27]and the LAW dataset for Devanagari [28]. They claimed that their technique outperforms other line-based techniques.

The work of [29] proposed a new faster approach using keyword spotting without the need for any language model for cursive English handwriting recognition. They used BLSTM (bidirectional long-short term memory) Neural Networks with sliding window approach. Each window has a one pixel width and it moves one pixel in each move from left to right through the text line. Nine features are extracted at each position. The extracted features are (0^{th}, 1^{st} and 2^{nd}) moment of the black pixels distribution, the position of the top-most and bottom-most black pixel, the inclination of the top and bottom contour, the number of vertical black/white transitions and the average gray scale value between the top-most and bottom-most black pixel. They used a dataset of 6,161 text lines. 1,000 labeled text lines and 5,161 unlabeled text lines. They chose 3,421 keywords to be spotted. Their system starts with available labeled data to train. Then, they select words from the set of high confidence recognized unlabeled data through several iterations. For four retraining iterations, they achieved an average recognition accuracy of 59.82% using one NN. Upon training 10 NNs, they achieved a recognition rate of 71%.

In [8], a method for spotting Arabic words using language models with gradient features and based on PAWs was proposed. Each PAW is normalized into 50 x 70 pixel image. They extracted the possible bi-grams, tri-grams and 4-grams for each word and built a language model. Their database consisted of 69 words that have been segmented into their PAWs. The number of words with one, two, three and four PAWs were 23, 23, 20 and 3, respectively. They reported 65% of recognition rate. One of the limitations of this work is the small size of their database. Another limitation is the inability of their method to classify words consisting of more than 4 PAWs since their language model does not exceeding 4-grams. Figure 2.1 shows Word Spotting Survey Summary.

Table 2.1: Word Spotting Survey Summary.



Year	Language	Document Type	Database	Features	Classifiers	Performance
2002[9]	English	Digital Library Of George Washington's manuscripts.	three sets of 10 consecutive pages	Length, area, aspect ratio, number of ascenders & descenders, and the number of black to white transitions.	Sum of squared differences, Euclidean distance mapping, XOR algorithm and Scott & Longuet Higgins (SLH) algorithm	73%
2003[10]	English	Digital Library Of George Washington's manuscripts.	Two different data sets of both 10 pages size. The first set is of Acceptable quality. While The second set is very degraded	projection profile, word profile and background/ink Transitions.	Dynamic Time Wrapping (DTW) classifier.	72%.
2004[11]	Arabic	historical documents Ottoman Empire archives	102 document images30 candidate query imagesin set, and 16 query imagesare tested	angular span, distance span and Wavelet domain feature	Euclidean distance and the histogram intersection values as similarity measures	86%.
2008[3]	Arabic	handwritten or printed historical documents.	123 historical document pages	loops, ascenders, descenders and diacritics, producing a symbolic representation for each connected component	Levenshtein distance	77.78%
2009[12]	Arabic	historical handwritten documents	training dataset consists of 200 words	structural and statistical features	Neural Networks	89.3%
2011[4]	Arabic	historical manuscripts	Two historical Arabic Manuscripts. 34 pages with 5500 words	concentric, rectangular, angular and concentric polar grid features	content based image retrieval (CBIR) techniques and latent semantic indexing (LSI)	78.7%.
2012[6]	Arabic	Handwritten	69 words	Gradient features based on PAWs	language models	65%

2.2 Handwriting Text Recognition

Here, we present selected research works in handwritten text recognition at the character and word levels for English, Persian and Arabic.

The work of [13] proposed a new recognition system for handwritten Arabic characters. They used robust noise independent algorithm with tree structure. They generated a set of rules in sequence to match a character tree to fuzzy constrained character graph models. One problem with this work is speed. Arabic handwritings of four writers were used as the main database in the training and testing. They tested their prototype on the four writers and got the following rates 100%, 97.4%, 94%, 99.4% for the first, second, third and fourth writer, respectively.

Broumandnia et. al, used M-band packet wavelet transform features and compared them to Invariant Zernike moments and Fourier wavelet features in recognizing handwritten words [30]. Rotation and scale invariant characteristics of M-Band Packet wavelet transform were used to extract features and the Mahalanobis distance was employed to classify and recognize the words. Their dataset consisted of 224 Arabic words that have been written by five different writers with 8 different orientations between 45 and 315 degrees. A recognition rate of 95.8% was reported.

Lawal et al. proposed a new method for Arabic numerals recognition [31]. Abductive network architecture was used as a classifier with a set of features such as using histograms of contour points and Freemans chain codes. The dataset consisted of 21120 samples of handwritten digits (Indian Numerals 0-9) that were written by 44 different writers. Each writer wrote 48 samples of each digit. 15840 samples were used for training and 5280 samples were used for testing. The results showed that increasing the number of features does not always give better performance. They reported a recognition rate of 99.03% using 32 features, while work of [32] reported a recognition rate of 97.99% using 120 features.

Bahmani et. al, proposed using radial basis functions based Neural Network (RBF) and Genetic algorithms for Arabic/Farsi handwritten word recognition [33]. They extracted wavelet coefficients from smoothed word images in four directions. Although they had a database consisting of 3300 images containing the handwriting of 30 Persian names, they chose to include only 100 samples in the training set and 10 samples for testing. They reported a recognition rate of 97% on that limited dataset.

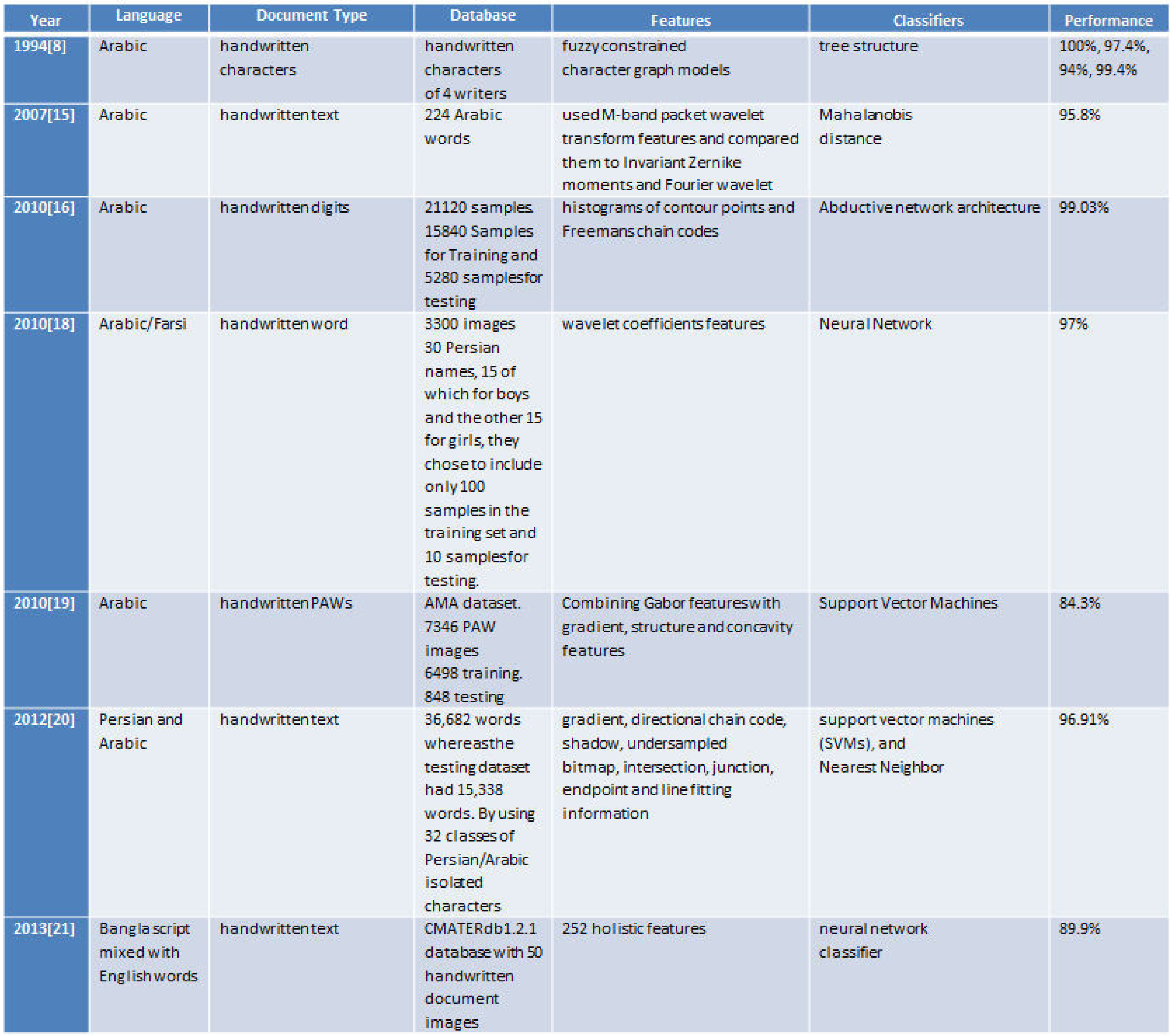
Combining Gabor features with gradient, structure and concavity features (GSC) in [34] gave better results than using each one of them separately on Support Vector Machines. The AMA Arabic dataset [27] was used as the main database in this work. This technique depended on 7346 PAW images from 34 most frequent PAW selected classes. The used Dataset consisted of 6498 training PAW images, and 848 testing PAW images. Best recognition rate of 84.3% was reported using GSC mixed with Gabor features [34].

Authors of [35] investigated different sets of features and employed different classifiers in the recognition of Persian and Arabic handwritten text. These features were computed based on gradient, directional chain code, shadow, undersampled bitmap, intersection, junction, endpoint and line fitting information. The classifiers were support vector machines (SVMs), 1-Nearest-Neighbor (NN), 3-Nearest Neighbor (3-NN) and 5-Nearest Neighbor (5-NN). The training dataset consisted of 36,682 words whereas the testing dataset had 15,338 words. By using 32 classes of Persian/Arabic isolated characters, the best reported recognition rate was 96.91% using 400 gradient features and the SVM classifier.

Acharyya et. al, proposed recognizing English words from their overall shape in carrying out handwritten text recognition [36]. They used neural network classifier with 252 holistic features, which were computed by dividing the whole word image into hierarchical vertical segments. The hierarchical segmentation process starts with depth 0 of 2^0 sections continuing up to depth-5 of 2^5 sections. At each section, the four longest run features are computed in four directions: row-wise east, column-wise north and diagonals of northeast & northwest. The longest run feature is computed by the sum of the lengths of the longest bars that fit consecutive black pixels. They carried out their experimentation on CMATERdb1.2.1 database [37] which consists of 50 handwritten document images written in Bangla script mixed with English words. They extracted word images manually, and the accuracy of their method was evaluated using three fold cross validation methods and 25000 iteration. Best recognition rate of 89.9% was reported.

Figure 2.2 shows Handwriting Recognition Survey Summary.

Table 2.2: Handwriting Recognition Survey Summary.



Year	Language	Document Type	Database	Features	Classifiers	Performance
1994[8]	Arabic	handwritten characters	handwritten characters of 4 writers	fuzzy constrained character graph models	tree structure	100%, 97.4%, 94%, 99.4%
2007[15]	Arabic	handwritten text	224 Arabic words	used M-band packet wavelet transform features and compared them to Invariant Zernike moments and Fourier wavelet	Mahalanobis distance	95.8%
2010[16]	Arabic	handwritten digits	21120 samples. 15840 Samples for Training and 5280 samples for testing	histograms of contour points and Freemans chain codes	Abductive network architecture	99.03%
2010[18]	Arabic/Farsi	handwritten word	3300 images 30 Persian names, 15 of which for boys and the other 15 for girls, they chose to include only 100 samples in the training set and 10 samples for testing.	wavelet coefficients features	Neural Network	97%
2010[19]	Arabic	handwritten PAWs	AMA dataset. 7346 PAW images 6498 training. 848 testing	Combining Gabor features with gradient, structure and concavity features	Support Vector Machines	84.3%
2012[20]	Persian and Arabic	handwritten text	36,682 words whereas the testing dataset had 15,338 words. By using 32 classes of Persian/Arabic isolated characters	gradient, directional chain code, shadow, undersampled bitmap, intersection, junction, endpoint and line fitting information	support vector machines (SVMs), and Nearest Neighbor	96.91%
2013[21]	Bangla script mixed with English words	handwritten text	CMATERdb1.2.1 database with 50 handwritten document images	252 holistic features	neural network classifier	89.9%

In addition to KHATT and IFN/ENIT databases, there are a number of Handwritten databases that were developed.

The work of [38] built an image database for Arabic handwritten character recognition. The database consists of 28000 images written by 100 native Arabic writers. Each writer wrote the Arabic letter 10 times in a form. They divided the scanned images into 80% for training and the rest for testing.

The work of [39] proposed text writer identification system on writer word images using K-nearest neighbor classifier. They used edge-direction distribution, moment invariants, and word measurements feature extraction. They built a database of offline Arabic handwriting text for writer identification research. It was written by 100 writers and a total of 32,000 text images. 75% of words were used in training and 25% were used for testing. They showed that longer Arabic words and phrases have higher impact on writer identification. They achieved greater than 90% of recognition rate of top ten writers for certain words. The difference in the obtained experimental details and the method of data collection make it difficult to compare it to other studies.

The work of [40] presented IFN/Farsi-Database similar to the idea of developing IFN/ENIT-Database for the Arabic handwritten words recognition. The database consisted of 7,271 binary images and comes from 1,080 Iranian Province/city names, written by 600 writers. Ground truth, number of characters, sub-words and dots are included for each word image.

The work of [41] introduced an Arabic handwritten text images database called (AHTID/MW) that includes 3710 text lines and 22896 words and written by 53 different writers. This database can be used for open vocabulary recognition, word segmentation and writer identification.

The work of [42] proposed an analytical approach and employed an off-line handwriting recognition system using HMM. They proved that the accuracy based on baseline features detection algorithm was higher. Features were based on the densities of foreground pixels and concavity of sliding window, reporting 74.90% recognition rate. Lower and upper baselines features were added to previous features, reporting 86.51% of recognition rate. Character models were used without character pre-segmentation using IFN/ENIT-Database of Tunisian handwritten country/village names.

Kchaou et. al, [43] presented comparative study to build good effective retrieval system. Segmentation and word spotting methods, i.e. characterization and matching for printed and handwritten Arabic texts, were done on a small Arabic dataset. They showed that the scale space is more reliable than horizontal and vertical profile features in segmentation. For characterization, geometrical moments take more time than projection profile. In matching, similarity cosine metric takes less time than DTW. They showed that vector combination is the most reliable one in term of performance.

The work of [44] produced an optical character recognition system using segmentation algorithms for Arabic handwritten characters. This algorithm depended on script height, character width, pen thickness and word/subword gaps. Used samples have very neat and very bad Arabic handwriting script. Fine segmentation gave lower accuracy compared to text-lines and coarse segmentations.

The work of [45] proposed active shape models for handwritten Arabic words and text pages. They used 28046 online samples for character synthesis and statistical properties. They used IESK Arabic database to simulate word slant, skew or baselines. In order to meet the properties of handwritings, they used rendering techniques and interpolation methods. They validate the segmentation algorithm on data to get comparable results.

Comparing to handwritten text recognition, printed text recognition reported high recognition rate such as the work of [46] focused on segmenting and identifying individual letters of printed text to use them in optical character recognition based on the form of letters (isolated, initial, medial and final). They used multiple grids of SIFT descriptors features. They used PATS and APTI datasets with 5 fonts. They reported 98.87% to 100% recognition rate for PATS and it was competitive for APTI.

CHAPTER 3

FRAMEWORK FOR PAW GENERATION AND RECOGNITION

Text recognition of handwritten Arabic is not an easy task since written characters, and subsequently PAWs, are not uniform even for the same writer. The process becomes more complicated when parts of words in different lines happen to be close to each other such that assigning each part to a line becomes a challenging task on its own. Another challenge in this regard is the existence of discontinuous curves, whether from the way a human writes, or due to preprocessing of images, e.g. binarization. Last but not least, some characters have very similar shapes and may only differ in the number of dots or diacritics. Arabic handwritten text recognition has been performed at the page, word, PAW and character levels [47] [48] [8] [13].

3.1 PAW Image Generation

In this thesis, we address handwritten text recognition at the PAW level, which has the advantage of avoiding character segmentation.

Many challenges exist in extracting PAWs from handwritten text. The most common are:

- Associating dots with their respective characters.
- Handling joined curves (two or more separate curves that intersect) and broken/split curves (one curve that appears consisting of two or more split curves).
- Dealing with large sizes and small sizes of identical text bodies.
- Coping with asymmetric character sizes (e.g. big size of some dots that may exceed the size of characters).

Since each word may consist of one or more connected components, these connected components need to be extracted and labeled with their corresponding text. For example, the word Assalaam السلام consists of three PAWs, viz., Alif ا, Asslaa سلا , Meem م. The mechanism of PAW extraction depends on the type of connectivity model used. There are two such models: 4-connection and 8-connection sides. In the 4-connection model, the neighboring pixels considered are those to the west, east, north and south of the current pixel. In the 8-connection model, pixels to the west, east, south, north, north east, north west, south east and south west of the current pixel are considered. The two models are shown in Figure 3.1.

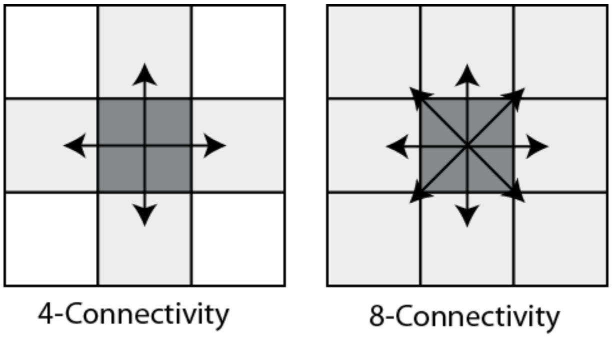


Figure 3.1: Four-Connectivity and Eight-Connectivity Models of neighboring pixels.

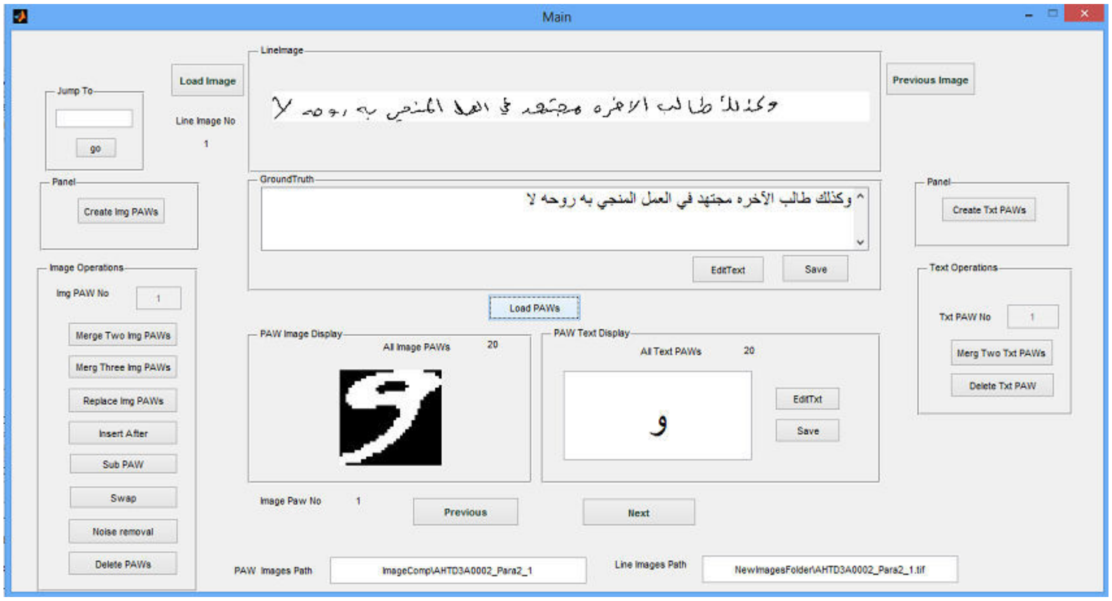
In order to extract, generate and annotate more accurate PAW images, we developed an algorithm to extract PAWs from line images. Although a lot of effort was done to fully automate this process, such goal was not achieved due to many challenges that accompany handwritten text. For example, one attempt was made to generate PAWs by finding the area containing black pixels, but found many cuts in the same region. We computed the minimum distance between every two consecutive regions and tried to differentiate between the main body of PAWs and dots/diacritics by determining a single threshold value, with no success. Hence, we developed a semi-automatic system, in which human intervention may be required. The system was developed in a way that facilitates the proper annotation of PAWs and the efficient correction of errors. Diacritics and dots have also been included in the bounding box of their corresponding PAW image. Table 3.1 outlines the PAW generation algorithm.

Table 3.1: Connected Components Generation Algorithm for Arabic Handwritten Text.

1. Convert the input line image into a binary image;
2. Read the line image from right to left;
3. Extract all connected components of the line image;
4. Arrange the generated connected components by area, from largest to smallest;
5. For each connected component do:
    - Crop Sub-Image for indexed Connected Component with its lower and upper parts on same Bounding Box.
    - Clear the main part of original binary image with dimension of generated Bounding Box.
    - If there are any black pixels in image repeat 5.
    - Otherwise stop.

Corresponding to each line image, there is text ground truth. Our system faced the challenge of matching each identified PAW with its corresponding text. Chapter 1 discussed the difficulties related to Arabic handwritten text recognition, while the most common challenges of PAWs extraction were discussed at the beginning of this chapter. In most cases, when the number of identified PAWs in a line image is equal to the number of connected components of the corresponding ground truth line text, such matching is straight forward and is, most of the time, correct. However, when these two values differ, human intervention becomes necessary. This forced us to develop utilities and functions that can correct these errors. These functions are divided into two parts. The first part was related to the application of corrections on the PAW images, whereas the second one is related to corrections of the corresponding ground truth text.

An easy to use graphical user interface was developed using matlab. It supports all operations that are needed to correct any errors in the PAW generation process. The system takes as input two directory names containing the line images and their corresponding ground truth text. Figure 3.2 shows the framework interface of PAWs generation. The next two sections illustrate the supported functionalities for correcting linage image PAWs and ground truth text, respectively.



Line Image No 1

GroundTruth: وكذلك طالب الأخره مجتهد في العمل المنجي به روحه لا

Img PAW No 1

All Image PAWs 20

All Text PAWs 20

و

Txt PAW No 1

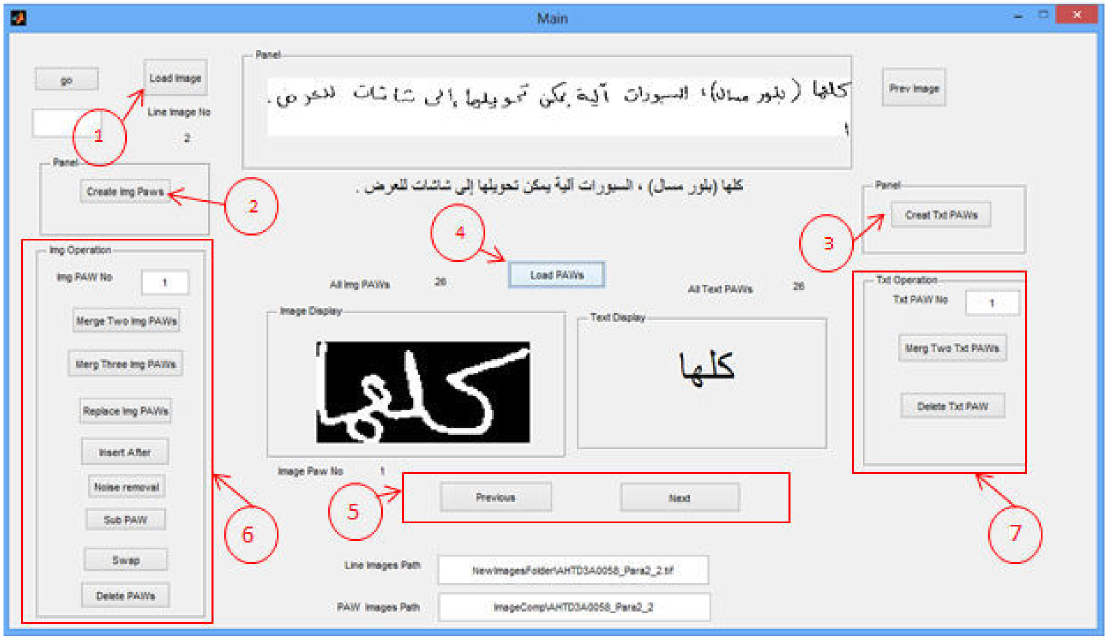
Image Paw No 1

PAW Images Path ImageComp\AHTD3A0002_Para2_1

Line Images Path NewImagesFolder\AHTD3A0002_Para2_1.tif

Figure 3.2: Framework Interface.

As initial information, the name of the line image is made the same as that of the corresponding ground truth file name. When we run our framework, the main interface appears as shown in Figure 3.3 with arrows and sequenced steps of framework usage. First, the line image is loaded with its ground truth. Second, create PAW images and keep them in image component folder with same name of the original line image. Third, create PAW text of line image ground truth and keep them in text component folder with same name of ground truth of line image. Fourth, load all PAWs of specific line image and its ground truth. Fifth, Move one by one among all PAWs by using next and previous buttons. Sixth, use all required operations to fix line image PAWs and finally, use all required operations to fix the ground truth PAWs.



كلها (بلور مسال) ، السورات آلية يمكن تحويلها إلى شاشات للعرض .

Line Image No 2

Img PAW No 1

All Img PAWs 28

All Text PAWs 28

Txt PAW No 1

Image Paw No 1

Line Images Path NewImagesFolder\AHTD3A0058_Para2_2.tif

PAW Images Path ImageComp\AHTD3A0058_Para2_2

Figure 3.3: Framework Interface Operations .

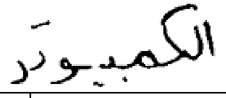
3.1.1 Correcting Line Image PAWs Functions

The following functions are used to handle any corrections necessary for generating proper PAW images from the line image.

- Merging two, three or four PAWs:

  This function allows the user to choose any two, three or four sequential PAWs, and then merge all of them into a single image PAW. This is very useful when errors in the binarization and/or the consideration of a diacritic or a dot as a PAW, due to its abnormally large size. In this case, there is a need to combine all of them into one single PAW image. Figure 3.2 shows the PAWs of word Alcomputer الكمبيوتر before and after merging.

Table 3.2: PAWs of word Alcomputer الكمبيوتر before and after merging.

word					
PAWs before merge					
PAWs after merge					

- PAW Replacement:

  This function allows the replacement of a selected PAW by index, with a manually generated one. Sometimes the PAW itself consists of more than two parts as cuts inside the curve that forces one to select the original PAW by using the mouse cursor to determine the bounding box of the selected PAW. Figure 3.4 shows the PAW llhusoo للحصو of word llhusool للحصول that consists of many cut curves. In this case, the manually generated PAW replaces the other automatically generated PAWs.

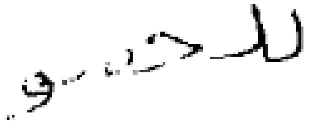


Figure 3.4: PAW Replacement Example.

- Insert After:

  This function allows to insert a PAW after the selected one by index. The inserted PAW is manually generated by determining the bounding box. a cursor

- PAW Splitting:

  This function divides the selected PAW into two different sequential PAWs by drawing bounding boxes around the newly generated PAWs manually. It is useful when two or three consecutive PAWs appear as one PAW. The user can, then, select the bounding box of the first PAW by using mouse cursor to split them. Figure 3.5 shows merged PAWs that can be split.

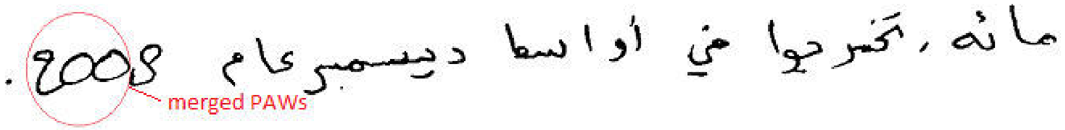


Figure 3.5: Line Image and PAW split Example.

- PAW Swapping:

  This function enables the user to swap any selected PAW with the next one. This function is useful when a sequence of more than one number appears through the handwritten text, whereas their text ground truth appear in reverse order. Figure 3.6 shows a sequence of numbers that need swapping to match their ground truth PAWs.

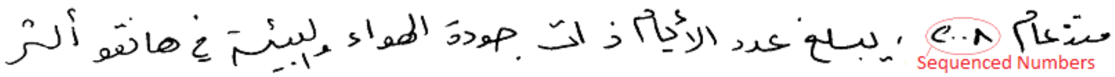


Figure 3.6: Line Image and PAW Swapping Example.

- Noise Removal:

  This function removes noise and/or scratches that are not related to the selected PAW image. It is manually removed by drawing a bounding box on the area that needs to be removed. Figure 3.7 shows the extra part under the word Fossol فصول that is removed.

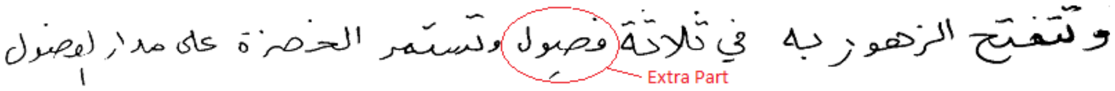
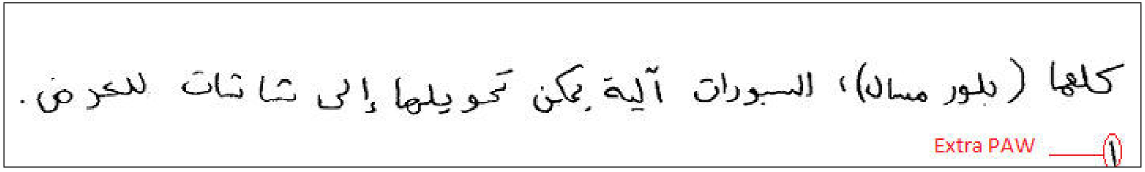


Figure 3.7: Extra Part Under Word Fossol فصول..

- Delete Image PAW:

  This deletes the selected PAW image from the folder of line image PAWs. We need this function when a PAW overlaps with another line image in the line image producing process. Figure 3.3 shows an extra PAW that must be removed.

Table 3.3: Extra PAW of line Image.


كلها (بلور مسال)، السبورات آلية يمكن تحويلها إلى شاشات للعرض.
Extra PAW

3.1.2 Correcting Ground Truth Text Functions

The existing ground truth of Arabic text for any handwritten database corresponds to the ideal handwriting of that text. Since this is not the case, some functionalities are needed in the system that will generate ground truth text that correctly corresponds to the PAW image it represents. The following functionalities were implemented in our system to correct ground truth text:

- Merge:

  This function enables the user to merge any two or more sequential text PAWs to match merged image PAW. Table 3.4 shows the merged PAWs in line image of word Behulul بحلول that must be merged in text PAWs.

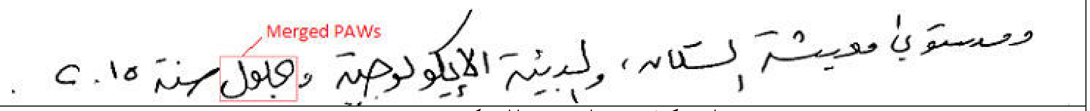
- Delete:

  This function is used to delete the selected text from the ground truth that does not correspond to any PAW image. For example, the text Meeladiah ميلادية. Table 3.4 shows an extra word Meeladiah ميلادية that doesn't exist in the original line image.

- Modify:

  This function is used to modify the content of a selected text PAW. It can be used to correct mistakes that exist in the original ground truth text. Table 3.4 shows the wrong word Shanah شنة that must be modified to Sanah سنة as in the original line image.

Table 3.4: Text Ground truth Operation

Line Image	 Merged PAWs
Ground truth	ومستوى معيشة السكان، والبيئة الإيكولوجية و بحلول شنة ٢٠١٥ ميلادية

3.2 Determining PAW Classes

After generating the PAW images and their ground truth, we need to address an important issue: How many PAW classes that need to be considered and what do they look like. One may be tempted to consider a PAW for each connected component we find. This strategy may not be best, as this means that there will be many PAW classes, most probably resulting in very poor performance. Instead, we group‘’similar” PAWs into a set of basic shapes (clustering). This is done by considering the main components of the PAWs regardless of any dots or diacritics.

For example, a single PAW may represent any of the following PAWs: Bayt بيت , Bent بنت , Nabt نبت , Tabt تبت. Hence, this PAW is encoded as ‘’xxz”, where the x may refer to any of the letters Baa بـ , Taa تـ , Thaa ثـ , Noon نـ , Yaa يـ and the z may refer to one of Baa ب Taa ت Thaa ث since the shapes of the other two letters in the end of the word are different, viz. Noon ن and Yaa ي. Figure 3.8 shows matching and clustering system and table 3.5 shows the combined model of Arabic letters that are similar in shapes depending on their positions in the beginning, middle, or end of a PAW, or if the PAW consists of a single character.

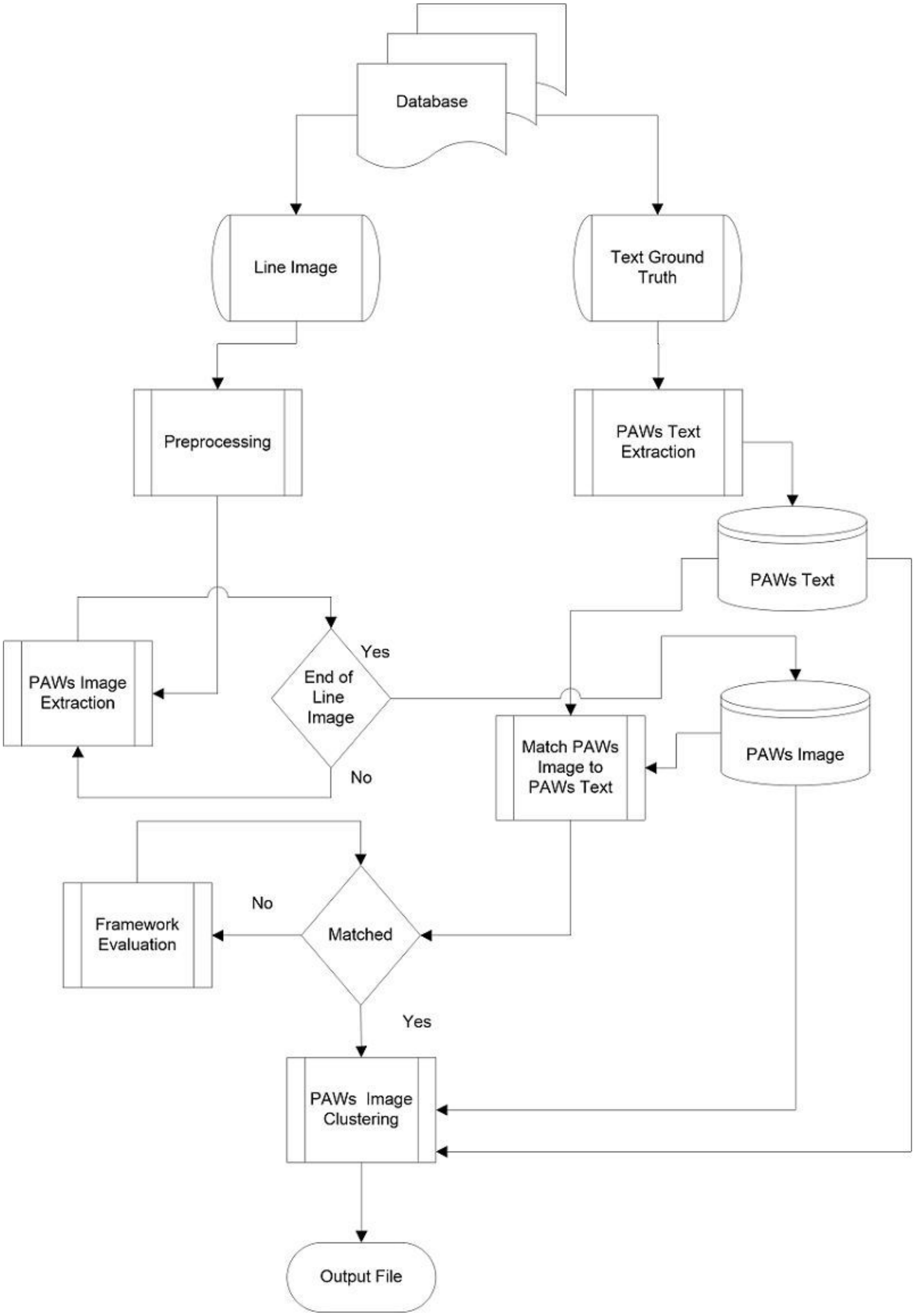


Figure 3.8: PAWs Matching and Clustering System.

Table 3.5: Combined Model of Arabic Letters

Combined Model			
beginning	Middle	End	Alone
			ء
		ا ـا أ ـأ إ ـإ آ ـآ	
ـبـ بـ ـتـ تـ ـثـ ثـ		ب ـب ت ـت ث ـث	
ـنـ نـ		ن ـن	
ـيـ يـ ـئـ ئـ		ي ، ئ ، ى	
ـجـ جـ ـحـ حـ ـخـ خـ		ج ـج ح ـح خ ـخ	
		د ـد ذ ـذ	
		ر ـر ز ـز	
ـسـ سـ ـشـ شـ		س ـس ش ـش	
ـصـ صـ ـضـ ضـ		ص ـص ض ـض	
ـطـ طـ ـظـ ظـ		ط ـط ظ ـظ	
عـ	ـعـ	ـع	ع
غـ	ـغـ	ـغ	غ
ـفـ فـ ـقـ قـ		ف ـف	
		ق ـق	
ـكـ كـ		ك ـك	
ـلـ لـ		ل ـل	
ـمـ مـ		م ـم	
هـ	ـهـ	ـه ـة	ه ة
		و ـو ؤ ـؤ	

3.2.1 PAW Class Encoding

In order to successfully use various software for feature extraction and classification, one needs to develop an encoding of PAWs so that it can be easily used thereof. Arabic letters forming each PAW class need to be converted to English letters. PAWs classes that are written in Arabic, e.g. Gaa قا - Ali علي - Mn من - Mohammed محمد, are split into their individual letters, viz., Gaaf ق, Alif ا - Aynع, Laam ل, Yaa ي - Meem م, Noon ن - Meem م, Haa ح, Meem م, Daal د. We used the encoding shown in Table 3.6 to encode PAW classes using the phoneme encoding for each Arabic letter. We added Underscore symbol after each character encoding representation to separate characters based on Arabic letter encoding table and separated underscore. The decoding process is easily implemented, which can convert them back to their original Arabic form. For example, Gaa قا will be Q_A , Ali علي will be AI_L_Y , Mn من will be M_N and Mohammed محمد will be M_HH_M_D.

Table 3.6: Arabic Letter Encoding.

Encoding	Arabic Letter	Encoding	Arabic Letter
E	Hamza ء	AI	Ayn ع
A	Alif ا	GH	Ghyan غ
B	Baa ب	F	Faa ف
T	Taa ت	Q	Gaaf ق
TH	Thaaث	K	Kaaf ك
ZH	Jeem ج	L	Laam ل
HH	Haa ح	M	Meem م
KH	Khaa خ	N	Noon ن
D	Daal د	W	Waw و
DH	Thaal ذ	H	Haa ه
R	Raa ر	Y	Yaa ي
Z	Zaay ز	AEU	Hamza over Alif أ
S	Seen س	AEL	Hamza below Alif إ
SH	Sheen ش	AAA	Mada over Alif آ
SS	Saad ص	YE	Hamza over Yaa ئ
DD	Daad ض	YA	Alif Magsoora ى
TT	TTaa ط	TU	Taa Marboota ة
DZ	Dhaa ظ	WU	Hamza over Waw ؤ

3.3 Feature Extraction

After the PAW images were generated, suitable features need to be extracted so that we can use them in the classification and recognition of PAWs. These features will be later used as input to the classification process. In this thesis, we studied two types of features. The first type is a set of four low-level features and the second type is Gabor features. All these features are represented as numerical values. The following discussion elaborates on the mechanism of feature extraction and presents brief summaries of their underlying theories.

3.3.1 Set of Four Low-Level Features

This set of features consist of four types of features that were studied in [7]. They include concentric circle features, angular line features, rectangular region features and circular polar grid features.

1. Concentric Circle Features:

   This feature is represented as a feature vector consisting of four features that are computed by counting the number of black pixels between concentric circles centered at the centroid of a PAW. The number of black pixels within a region in the PAW image is normalized by the total count of black pixels in the PAW image. Figure 3.9 shows the concentric circle feature.

the centroid of the image, C(cx,cy), is computed as follows.

$$r \times 1, r \times 2, r \times (n-1), r \times n \tag{3.1}$$

Where

- n is the number of concentric circles.
- r is concentric circles centered radius.

The distance between centroid of the PAW image and a given pixel by

$$D = \sqrt{cx^2 + cy^2} \tag{3.2}$$

The radius of each concentric circle is computed by:

$$r_i = \frac{D}{n} \times i \tag{3.3}$$

Where:

- i=1,2....n
- n is the number of concentric circles

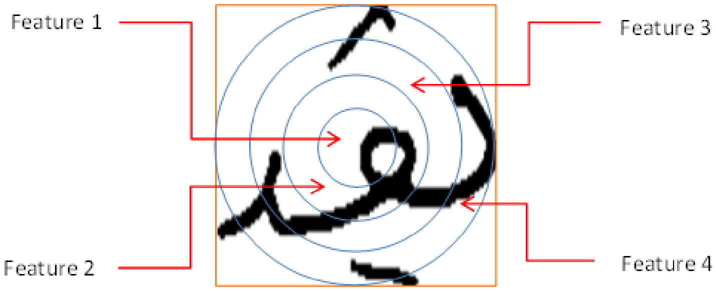


Figure 3.9: Concentric Circle Features.

2. Angular Line Features:

   This feature consists of eight scalar values that are computed by counting the number of black pixels in each sub-word image. The sub-word images are generated by dividing the image using 45 degree slices measure with respect to the horizontal axis of the centroid. The eight values are normalized by the total number of black pixels of the PAW image. Figure 3.10 shows the extraction of Angular Line Features.

   The centroid of image is C(cx,cy) was computed. Slope of two lines at θ=0 , 45 is computed by:

$$m = tan(\theta) \tag{3.4}$$

   computing centered at centroid of C(x,y)by

$$y = mx + b \tag{3.5}$$

For first slope:

$$y_1 = m_1 x + b_1 \tag{3.6}$$

For second slope:

$$y_2 = m_2 x + b_2 \tag{3.7}$$

Where b computed by

$$b_i = \frac{cx - cy}{m_i} \tag{3.8}$$

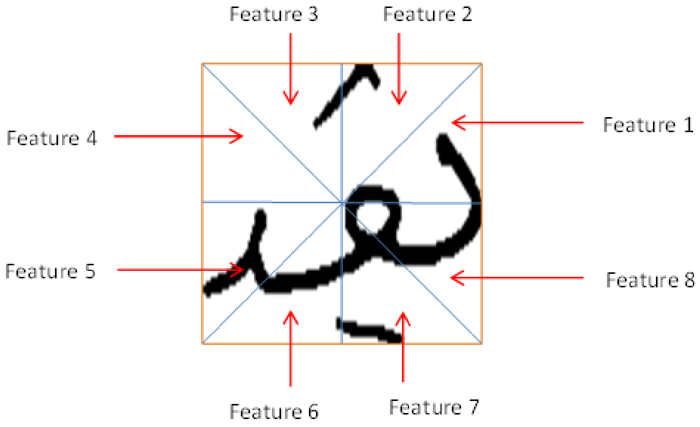


Figure 3.10: Angular Line Features.

3. Rectangular Region Features:

Rectangular region feature set consists of nine features that are generated by computing the number of black pixels in each of the nine rectangular regions. The value for each region is then normalized, as explained previously. Figure 3.11 shows the extraction of rectangular region features. To compute width and hight of each rectangular region by:

$$RegionWidth = \frac{ImageWidth}{3} \tag{3.9}$$

$$RegionHight = \frac{ImageHight}{3} \tag{3.10}$$

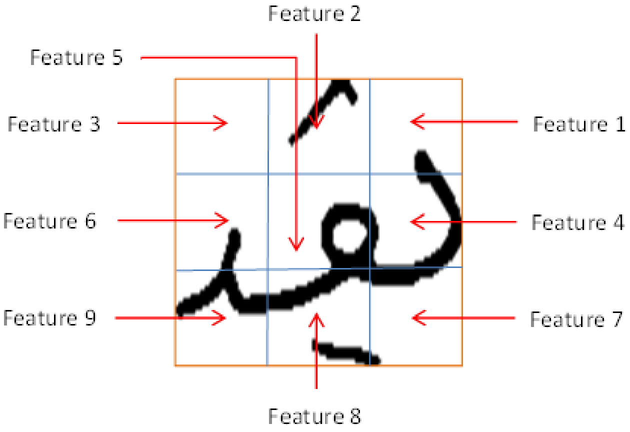


Figure 3.11: Rectangular Region Features.

4. Circular Polar Grid Features:/par This set consists of thirty two scalar values. It divides the PAW image into four concentric circles and four angular lines with equally spaced angles 0,45,90 and 135 degrees. The number of black pixels in each region is computed and then normalized. Figure 3.12 shows the extraction of circular polar Grid features

   We computed the slopes of θ =0, 45, 90 and 135 for the angular line features and used the same steps of the concentric circles for $n = 4$.

Where:

- n is the number of concentric circles.

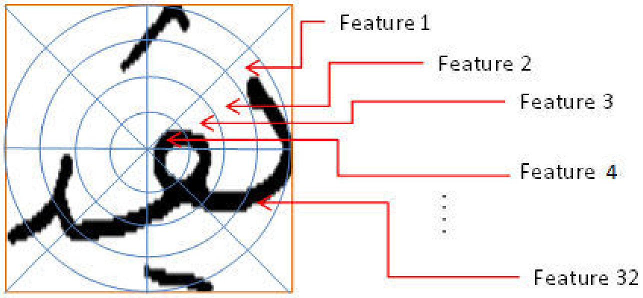


Figure 3.12: Circular Polar Grid Features.

3.3.2 Gabor Features

Gabor features have been used extensively in image processing. They have been utilized in character recognition as they have proved to be robust against noise and they do not require binarization [49]. They have also outperformed a set of gradient, concavity and structural features [34]. This has motivated us to investigate their use in PAW recognition.

The real part and imaginary part of Gabor functions are modulated in the space domain by a low pass Gaussian function [50] and can be formulated as:

$$g_e(x,y;\lambda,\theta) = e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})}\cos(2\pi\frac{x}{\lambda}) \tag{3.11}$$

$$g_0(x,y;\lambda,\theta) = e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})}\sin(2\pi\frac{x}{\lambda}) \tag{3.12}$$

Where:

λ is Gabor filter wavelength, θ is orientation angle, σ_x and σ_y control the spread of Gaussian window bandwidth.

Using the previous two equations on image I(x,y), the filter can be calculated by:

$$G_e(x,y;\lambda,\theta) = g_e(x,y;\lambda,\theta) * I(x,y) \tag{3.13}$$

$$G_0(x,y;\lambda,\theta) = g_0(x,y;\lambda,\theta) * I(x,y) \tag{3.14}$$

$$G_a(x,y;\lambda,\theta) = \sqrt{G_e^2(x,y;\lambda,\theta) + G_0^2(x,y;\lambda,\theta)} \tag{3.15}$$

Orientations can be calculated by:

$$\theta_k = \frac{2\pi k}{n}, k = (0, 1, 2, ..n - 1) \qquad (3.16)$$

Where:

n is the number of orientations.

We apply five different frequencies and eight different orientations to each PAW image. The mean and variance of Gabor filters are considered in our work. Therefore, each Gabor feature vector in our experiments consists of 80 feature values with 8 orientations and 5 scales. Figure 3.13 shows the real part of Gabor features for PAW Mn من while Figure 3.14 shows the magnitude of Gabor features for PAW Mn من. The images are so small and hence they seem covering no information

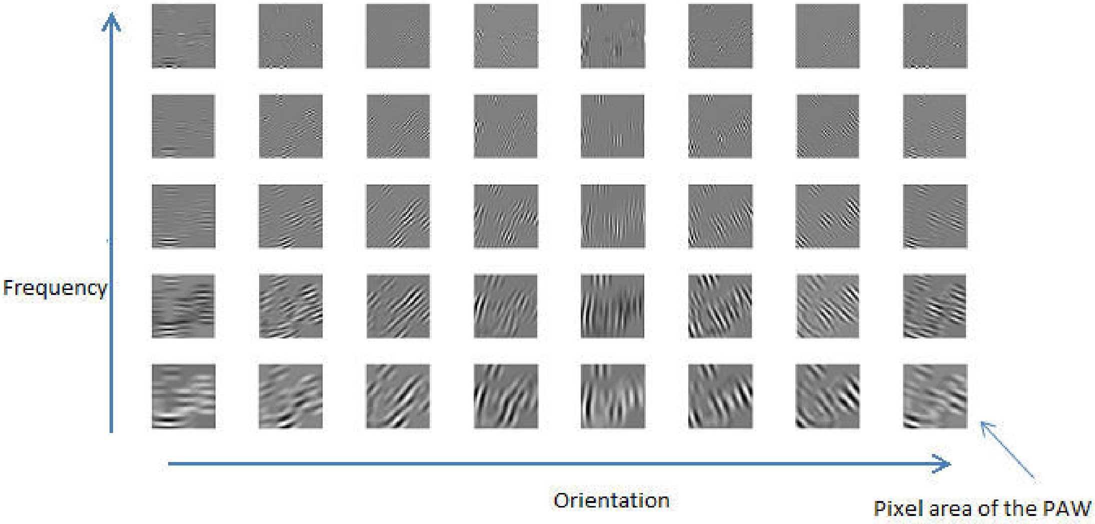


Figure 3.13: Real Parts Of Gabor Features For PAW Mn من.

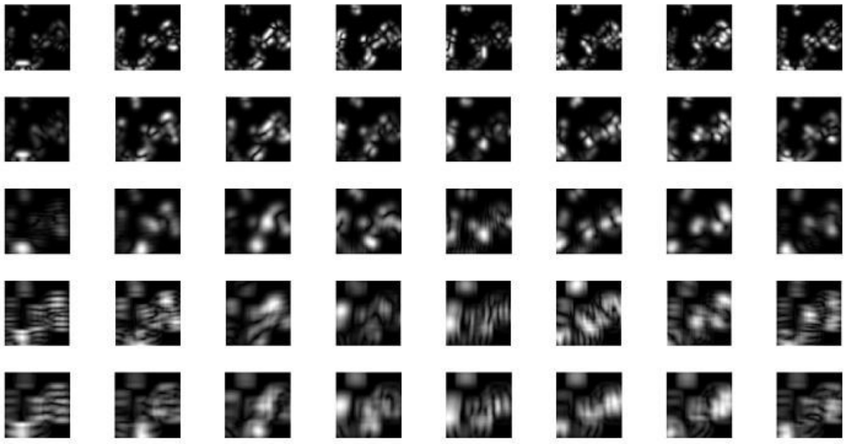


Figure 3.14: Magnitudes Of Gabor Features For PAW Mn من.

3.4 PAW Recognition

After determining PAW classes and the features that need to be examined, we need to employ a PAW recognition technique for input PAW images, as shown in Figure 3.15. In this work, we utilized two classification techniques to carry out PAW recognition: k-nearest neightbor (k-NN) and support vector machines (SVM). The main motivation behind using them is the existance of a large number of PAW classes. This means that the classification technique needs to be relatively fast in training and recognition. In addition, SVMs proved to be very successful in many classification problems.

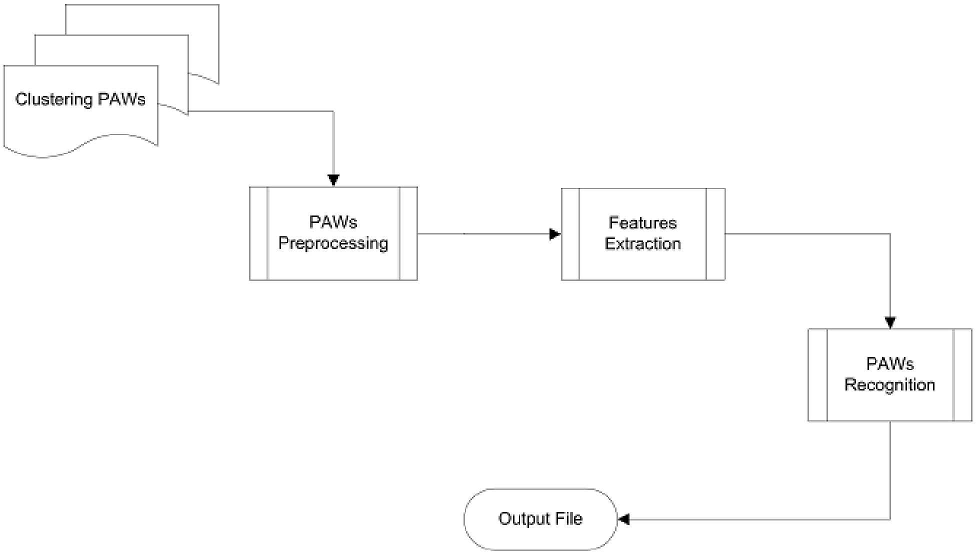


Figure 3.15: PAW Classes Recognition System.

3.4.1 K-Nearest Neighbor Classifier

Nearest Neighbor classifier is one of the most straightforward classifiers in machine learning techniques and is based on distance. It is used to identify the nearest neighbor to the test sample. When considering more than one neighbor for a tested sample to determine the class, this becomes k-nearest neighbor. The incentive to use it is the simplicity and ease of implementation [51].

In our work of PAW recognition, the distance is computed as follows:

$$|W_i - X| = \sqrt{\sum_{j=1}^{n}(w_{ij} - x_j))^2} \tag{3.17}$$

Where:

- $|W_i - X|$ is the distance between the feature vector of test instance w_i and the feature vectors of all instances in the training set X.

- n is the number of features.

- w_{ij} is the j^{th} feature of the feature vector of instance i.

- x_j is the j^{th} feature of the feature vector of the nearest test instance.

3.4.2 Support Vector Machines Classifier

The support vector machine classifier is a supervised machine learning algorithm that was originally proposed to carry out binary classification, i.e. data with two classes only. It achieves that by searching for an optimal hyper-plane that separates between the two classes. Figure 3.16 shows the hyperplane and the margin of the SVM Algorithm. Let $\mathcal{D} = \{(X_i, y_i)|1 \leq i \leq n\}$ consists of n pairs, where $X_i \in \Re^p$ represent the training data points in p-dimensional space, and $y_i \in \{-1, 1\}$ indicates the class to which Point X_i belongs.

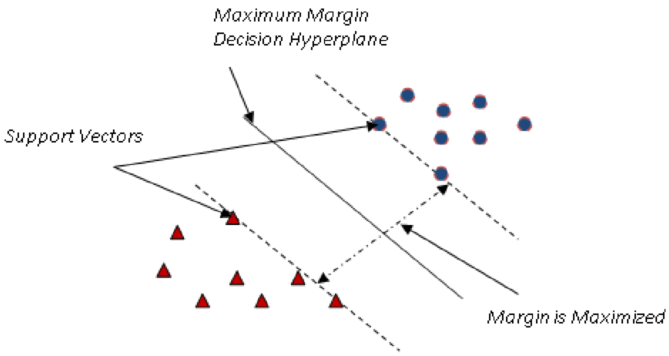


Figure 3.16: The Support Vectors and the Margin of the Classifier.

In order for the separating hyper-plane $\boldsymbol{w}$ to correctly classify all points, it should satisfy

$$y_i(\boldsymbol{w}^T X_i + b) \geq 1 \tag{3.18}$$

for all points, where b repersents the bias. Hence, the optimal separating hyper-plane can be computed by solving the following optimization problem:

$$\begin{aligned} &\text{Minimize} \quad \tfrac{1}{2}||\boldsymbol{w}||^2 \\ &\text{subject to} \quad y_i(\boldsymbol{w}^T X_i + b) \geq 1, \quad 1 \leq i \leq n \end{aligned}$$

Using the Lagrangian and quadratic programming (QP) for the nonlinear case [52], this problem reduces to:

$$\begin{aligned} &\text{Maximize} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j X_i^T Xj \\ &\text{subject to} \quad C \geq \alpha_i \geq 0, \ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

Where

- α_i's are the Lagrange multipliers,

- C is a tradeoff parameter between the error and the margin

Replacing the dot product above with a general kernel function K, we get

$$\begin{aligned} &\text{Maximize} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j K(X_i, Xj) \\ &\text{subject to} \quad C \geq \alpha_i \geq 0, \ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

Using the training data, we compute the values of α_i, where most of which are zero except for those associated with the support vectors. Now, to classify a test instance vector Z, we apply:

$$f(Z) = \sum_{j=1}^{N} \alpha_{t_j} y_{t_j} K(X_{t_j}, Z) + b \tag{3.19}$$

Where:

- N is the number of support vectors,
- X_{t_j} is a support vector in class t_j.

The most common Kernels of SVM are the linear kernel and the polynomial kernel. The equation for the linear kernel is:

$$k(X,Y) = X^TY + c \tag{3.20}$$

And the equation describing the polynomial kernel is

$$k(X,Y) = \left(\alpha X^TY + c\right)^d \tag{3.21}$$

Where

- α is the slope.
- c is a constant term.
- d is the degree of the polynomial.

Binary classification can be extended to multi-class classification using different approaches [53]. One commonly used way of implementing multi-class classification is the use of one-versus-rest classifier for each class, with a total number of classifiers equal to the number of classes. In this case, the margin that is generated by a given sample is computed for all classifiers. The sample is then assigned to the class of the classifier generating the biggest margin. Another strategy is to develop a set of one-versus-one classifiers, and then using majority vote to determine the class.

In our work, we choose WEKA (Waikato Environment for Knowledge Analysis) as the classification software to carry out our experiments. It is a professional classification tool that is supported and published by the University of Waikato New of Zealand [54] [55]. It adopts the one-versus-one strategy using John Platt's sequential minimal optimization algorithm for training the classifier. It either uses polynomial or Gaussian kernels. The SMO parameters used were the default parameters, viz.,:

$c = 1.0$, $\epsilon = 1.0E - 12$, kernel = PolyKernel; randomSeed = 1.

CHAPTER 4

DEVELOPMENT OF PAW DATABASE

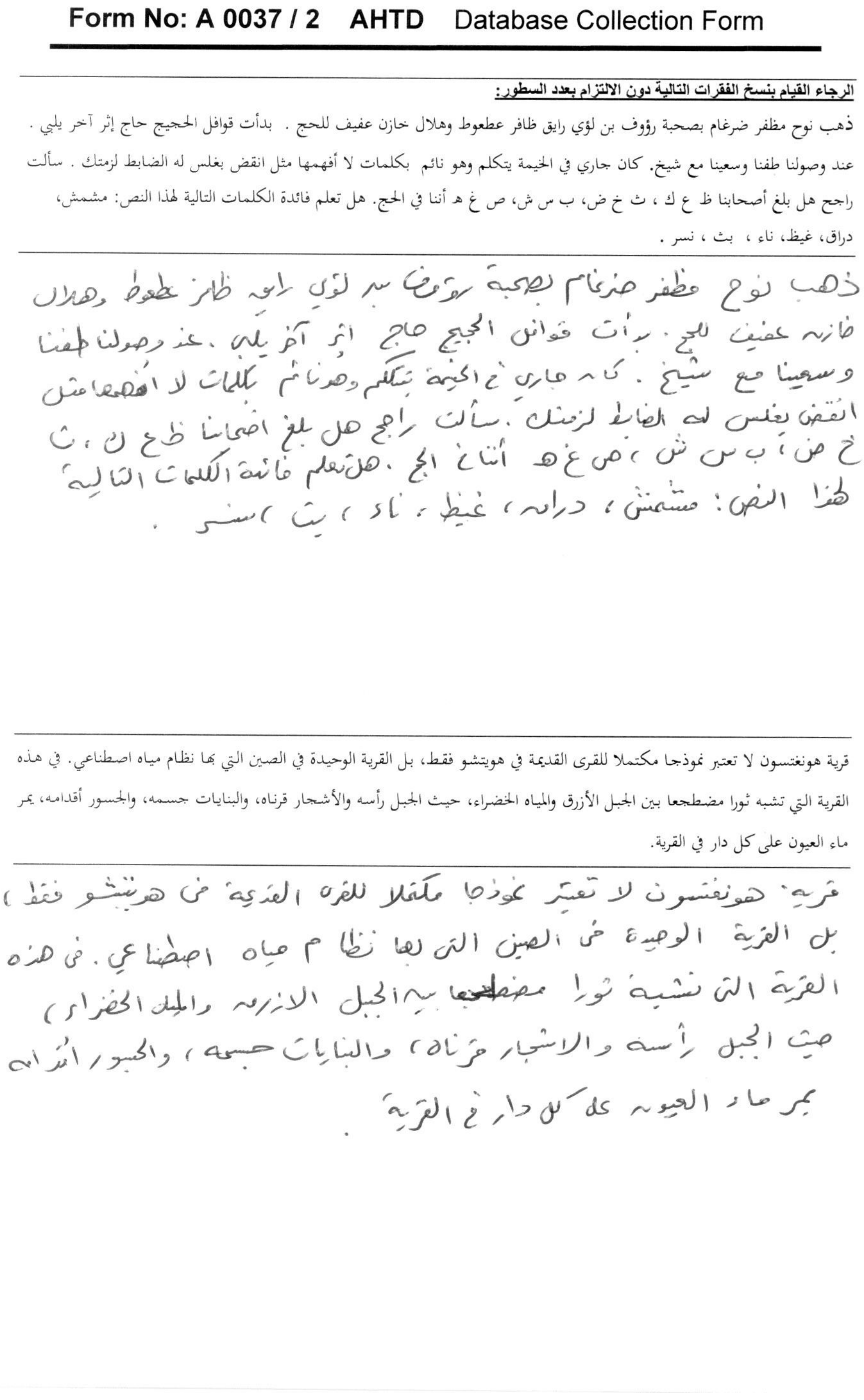
An important element for the success of most of the researches is a database that contains a large number of data. In this research, we need a large number of PAW data in the form of images and containing its corresponding text that was handwritten. In order to do this, we decided to use the available data in two well-known Arabic handwritten text databases, viz., KHATT and IFN/ENIT. A more elaborate description of the two databases is given below.

4.1 KHATT database

KHATT is an open Arabic offline database containing images of handwritten text. This database consists of 1000 forms that were written by 1000 different writers coming from 18 different countries. Each form consists of four predetermined paragraphs and two free form paragraphs. Two of the four paragraphs (Paragraph

1 and Paragraph 4) are identical in all forms, containing the same fixed Arabic text that includes all Arabic character shapes. Paragraphs 2 and 3 are different in each form, for a total of 2000 distinct paragraphs. These paragraphs were collected from 46 different sources, covering 11 different subjects. The forms were scanned at different resolution levels (200, 300 and 600 dpi), and were divided into 700 forms for training, 150 forms for validation and 150 forms for testing [2].

Figures 4.1, 4.2 and 4.3 show Paragraphs 1 and 2, Paragraphs 3 and 4, and Paragraphs 5 and 6, respectively. KHATT-Database also provides pre-segmented line images that we will use in our experimentation. Figure 4.4 shows a sample line image.



Form No: A 0037 / 2   AHTD   Database Collection Form

الرجاء القيام بنسخ الفقرات التالية دون الالتزام بعدد السطور:

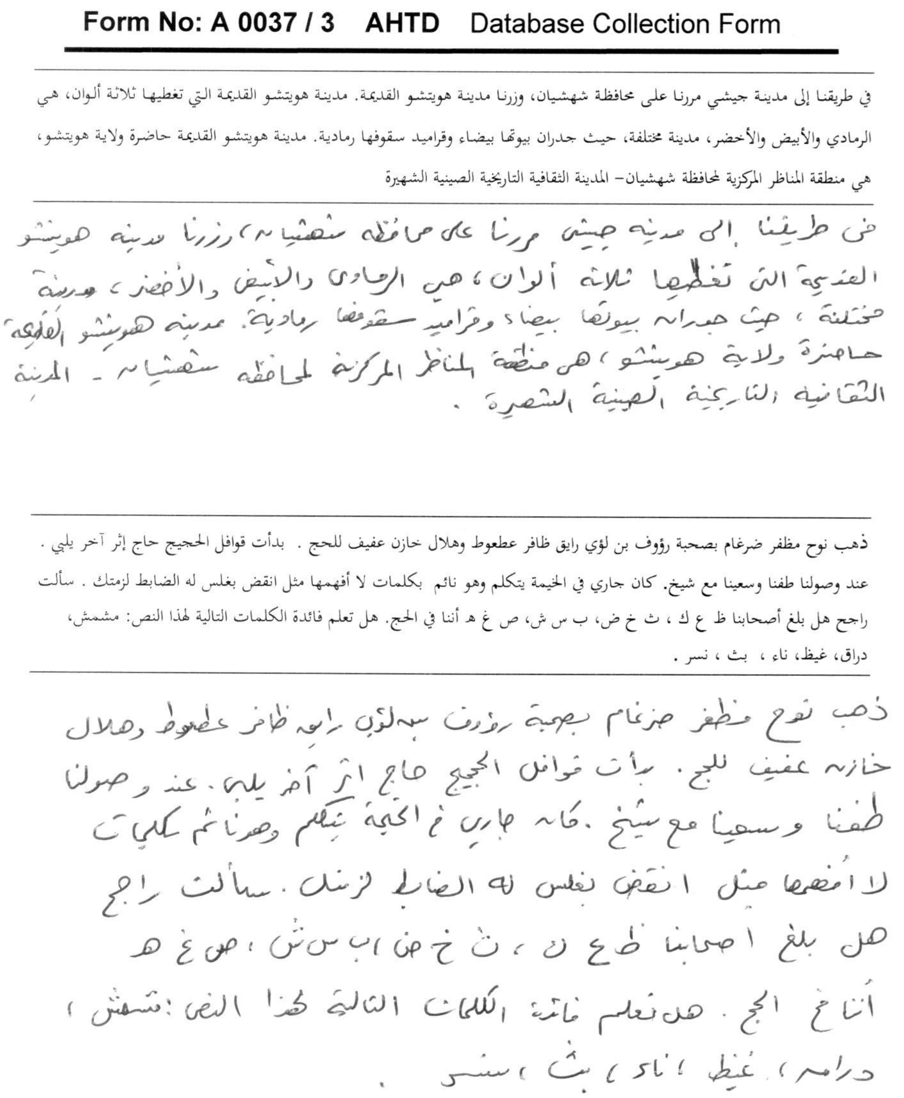
ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رايق ظافر عطعوط وهلال حازن عفيف للحج . بدأت قوافل الحجيج حاج إثر آخر يلبي .
عند وصولنا طفنا وسعينا مع شيخ. كان جاري في الخيمة يتكلم وهو نائم بكلمات لا أفهمها مثل انقض بغلس له الضابط لزمتك . سألت
راجح هل بلغ أصحابنا ظ ع ك ، ث خ ض، ب س ش، ص غ ه أننا في الحج. هل تعلم فائدة الكلمات التالية لهذا النص: مشمش،
دراق، غيظ، ناء ، بث ، نسر .

ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رايق ظافر عطعوط وهلال
حازن عفيف للحج . بدأت قوافل الحجيج حاج إثر آخر يلبي . عند وصولنا طفنا
وسعينا مع شيخ . كان جاري في الخيمة يتكلم وهو نائم بكلمات لا أفهمها مثل
انقض بغلس له الضابط لزمتك . سألت راجح هل بلغ أصحابنا ظ ع ك ، ث
خ ض ، ب س ش ، ص غ ه أننا في الحج . هل تعلم فائدة الكلمات التالية
لهذا النص : مشمش ، دراق ، غيظ ، ناء ، بث ، نسر .

قرية هونغتسون لا تعتبر نموذجا مكتملا للقرى القديمة في هويتشو فقط، بل القرية الوحيدة في الصين التي بها نظام مياه اصطناعي. في هذه
القرية التي تشبه ثورا مضطجعا بين الجبل الأزرق والمياه الخضراء، حيث الجبل رأسه والأشجار قرناه، والبنايات جسمه، والجسور أقدامه، يمر
ماء العيون على كل دار في القرية.

قرية هونغتسون لا تعتبر نموذجا مكتملا للقرى القديمة في هويتشو فقط ،
بل القرية الوحيدة في الصين التي بها نظام مياه اصطناعي . في هذه
القرية التي تشبه ثورا مضطجعا بين الجبل الأزرق والمياه الخضراء ،
حيث الجبل رأسه والأشجار قرناه ، والبنايات جسمه ، والجسور أقدامه
يمر ماء العيون على كل دار في القرية .

Figure 4.1: Paragraphs 1 and 2 of KHATT-Database.



Form No: A 0037 / 3   AHTD   Database Collection Form

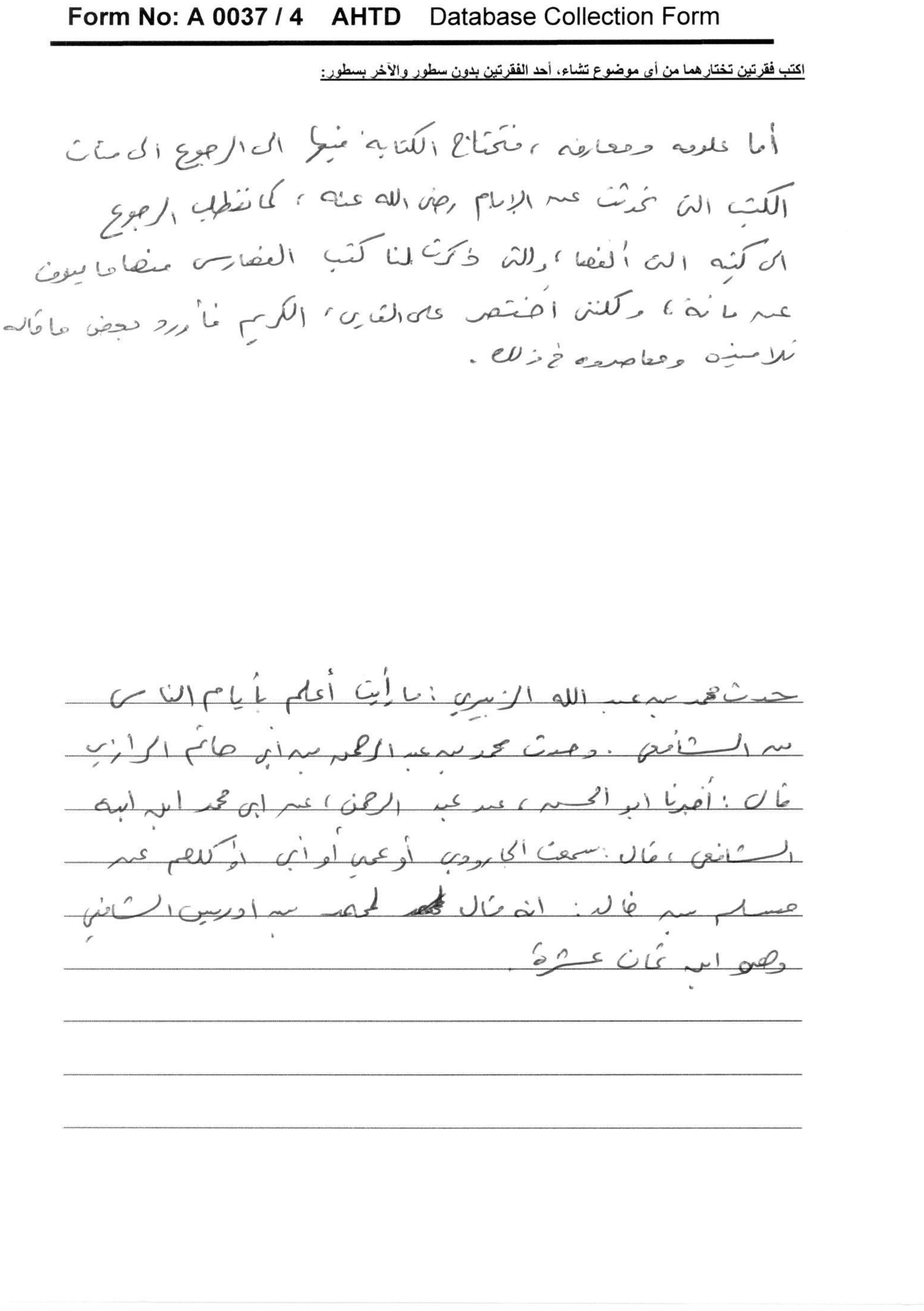
في طريقنا إلى مدينة جيشي مررنا على محافظة شهشيان، وزرنا مدينة هويتشو القديمة. مدينة هويتشو القديمة التي تغطيها ثلاثة ألوان، هي
الرمادي والأبيض والأخضر، مدينة مختلفة، حيث جدران بيوتها بيضاء وقراميد سقوفها رمادية. مدينة هويتشو القديمة حاضرة ولاية هويتشو،
هي منطقة المناظر المركزية لمحافظة شهشيان– المدينة الثقافية التاريخية الصينية الشهيرة

فى طريقنا الى مدينة جيشى مررنا على محافظة شهشيان، وزرنا مدينة هويتشو
القديمة التى تغطيها ثلاثة ألوان، هي الرمادى والأبيض والأخضر، مدينة
مختلفة، حيث جدران بيوتها بيضاء وقراميد سقوفها رمادية. مدينة هويتشو القديمة
حاضرة ولاية هويتشو، هى منطقة المناظر المركزية لمحافظة شهشيان - المدينة
الثقافية التاريخية الصينية الشهيرة.

ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رايق ظافر عطعوط وهلال خازن عفيف للحج . بدأت قوافل الحجيج حاج إثر آخر يلبي .
عند وصولنا طفنا وسعينا مع شيخ. كان جاري في الخيمة يتكلم وهو نائم بكلمات لا أفهمها مثل انقض بغلس له الضابط لزمتك . سألت
راجح هل بلغ أصحابنا ظ ع ك ، ث خ ض، ب س ش، ص غ ه أننا في الحج. هل تعلم فائدة الكلمات التالية لهذا النص: مشمش،
دراق، غيظ، ناء ، بث ، نسر .

ذهب نوح مظفر ضرغام بصحبة رؤوف بن لؤي رايق ظافر عطعوط وهلال
خازن عفيف للحج. بدأت قوافل الحجيج حاج اثر آخر يلبي. عند وصولنا
طفنا وسعينا مع شيخ. كان جاري فى الخيمة يتكلم وهو نائم بكلمات
لا أفهمها مثل انقض بغلس له الضابط لزمتك. سألت راجح
هل بلغ أصحابنا ظ ع ك ، ث خ ض، ب س ش، ص غ ه
أننا فى الحج. هل تعلم فائدة الكلمات التالية لهذا النص: مشمش،
دراق، غيظ، ناء، بث، نسر .

Figure 4.2: Paragraphs 3 and 4 of KHATT-Database.



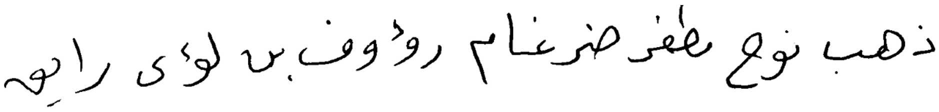
Form No: A 0037 / 4 AHTD Database Collection Form

اكتب فقرتين تختارهما من أي موضوع تشاء، أحد الفقرتين بدون سطور والآخر بسطور:

أما علومه ومعارفه، فتحتاج الكتابة فيها الى الرجوع الى مئات
الكتب التي تحدثت عن الإمام رضي الله عنه، كما تتطلب الرجوع
الى كتبه التي ألفها، والتي ذكرت لنا كتب الفهارس منها ما يربو
عن مائة، ولكنني أقتصر على القارئ الكريم فأورد بعض ما قاله
تلاميذه ومعاصروه في ذلك.

حدث محمد بن عبد الله الزبيري: ما رأيت أعلم بأيام الناس
من الشافعي. وحدث محمد بن عبد الرحمن بن أبي حاتم الرازي
قال: أخبرنا أبو الحسن، عن عبد الرحمن، عن ابي محمد ابن ابنة
الشافعي، قال: سمعت الجارودي أو عمي أو أبي أو كلهم عن
مسلم بن خالد: انه قال لمحمد بن ادريس الشافعي
وهو ابن ثمان عشرة.

Figure 4.3: Paragraphs 5 and 6 of KHATT-Database.

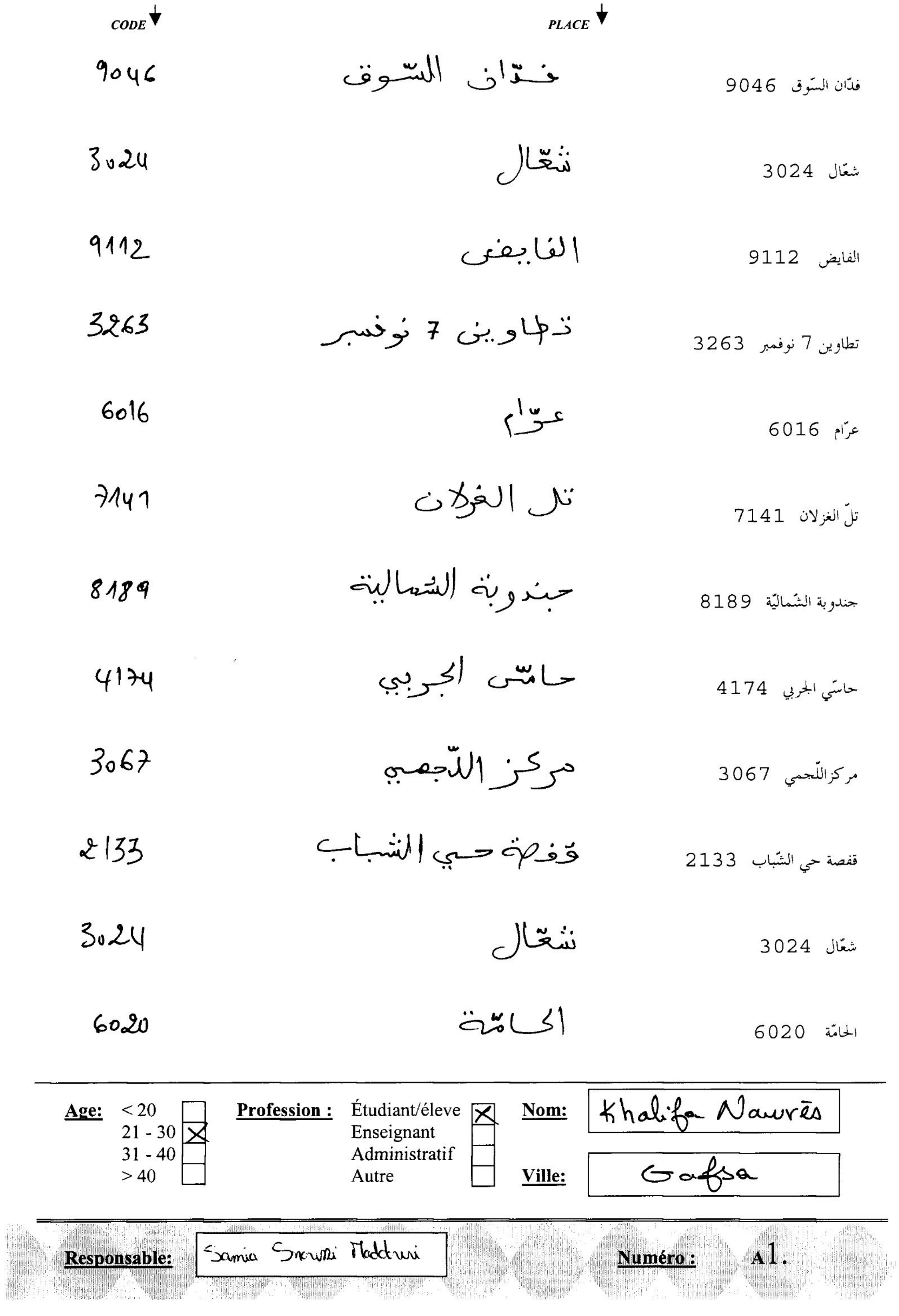


ذهب نوح ظفر ضرغام رؤوف بن لؤي رابح

Figure 4.4: Line Image of KHATT-Database.

4.2 IFN/ENIT-database

IFN/ENIT-Database is an Arabic handwritten database containing Arabic town/village names. It is also supported by ground truth information. It has 26400 names, generating 115585 PAWs and almost 210000 characters, that were written by 411 writers. There are five forms written by each writer using 60 names. A filled form of that database contains 12 printed Tunisian town/village names with its postal code and place in handwritten text. Information on the age, profession, and identity of the writer are at the bottom block of each form. It is a binary image scanned using 300 dpi [1]. Figure 4.5 shows IFN/ENIT-database Form and Figure 4.6 shows a Tunisian town/village name example.



CODE ▼	PLACE ▼	
9046	فدّان السّوق	9046 فدّان السّوق
3024	شعّال	3024 شعّال
9112	الفايض	9112 الفايض
3263	تطاوين 7 نوفمبر	3263 تطاوين 7 نوفمبر
6016	عرّام	6016 عرّام
7141	تل الغزلان	7141 تلّ الغزلان
8189	جندوبة الشمالية	8189 جندوبة الشّماليّة
4174	حاسّي الجربي	4174 حاسّي الجربي
3067	مركز اللّجمي	3067 مركزاللّجمي
2133	قفصة حي الشباب	2133 قفصة حي الشّباب
3024	شعّال	3024 شعّال
6020	الحامّة	6020 الحامّة

Age: < 20 □
21 - 30 ☒
31 - 40 □
> 40 □

Profession : Étudiant/éleve ☒
Enseignant □
Administratif □
Autre □

Nom: Khalifa Nawrēs

Ville: Gafsa

Responsable: Samia Snoussi Maddouri

Numéro : A1.

Figure 4.5: IFN/ENIT-database Form.

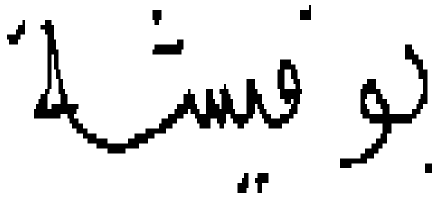


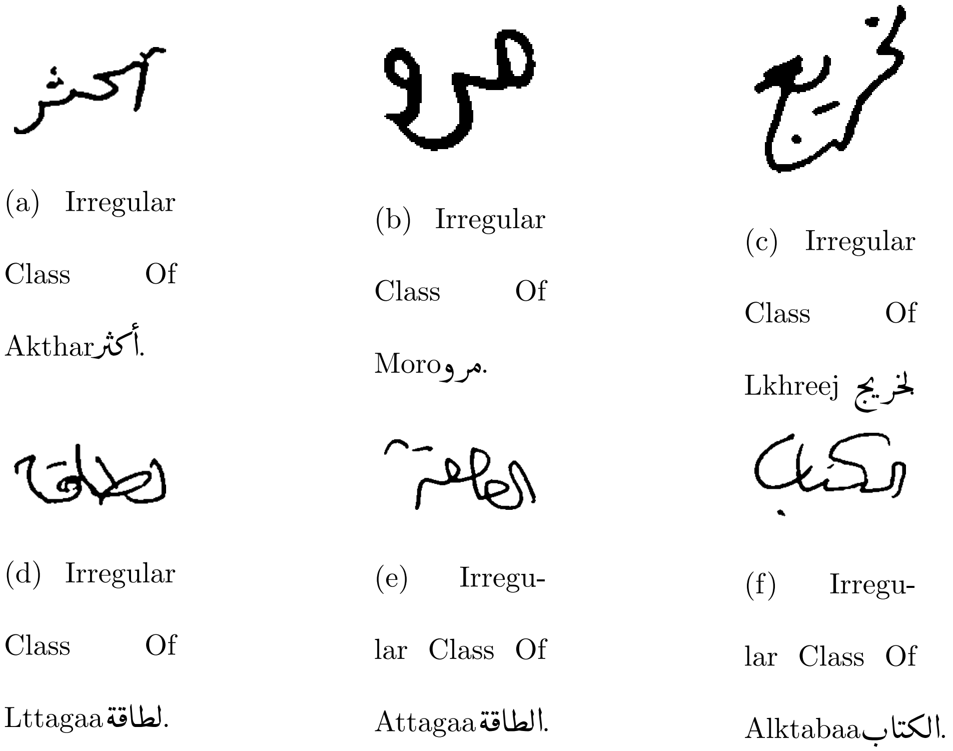
Figure 4.6: Tunisian town/village Name.

From the previous information of the two databases, we can see a very important point. KHATT database contains a huge number of line images and is an open-vocabulary database, whereas IFN/ENIT-database is closed vocabulary, restricted to Tunisian town/village names. Regardless of the used features, a closed database of limited number of words may give many training PAW Images of the same PAW and this can positively affect the recognition of tested PAWs. On the other hand, an open database is not specific to any special topic or area, which may lead to much fewer training PAW data of a specific PAW, and hence can negatively affect the PAW recognition.

Generation of PAWs is not a simple task to perform for Arabic handwritten text. Because, Arabic handwritten text is always cursive taking into consideration the difficulties that were explained in Chapter 1 and the challenges that were described in Chapter 3.

4.3 PAWs Database

Both databases were utilized to build our training and testing PAWs model. For KHATT-Database, we used the unique line images to extract PAWS. Two folders for training and testing were chosen. The number of Line images, number of extracted PAWs of those line images, the number of classes, the number of filtered classes and the used instances of each database in training or testing modes were collected. By filtered classes, we mean those classes that are produced after the irregular classes were kept out. An irregular class is the one with two or more consecutively connected PAWs that should have be separated, but are in fact continuous. Examples of such include Akthar أكثر , Moro مرو , Lkhreej لخريج , Lttagaa لطاقة , Attagaa الطاقة , Alktaabaa الكتاب which should have been Alif أ _ Kthar كثر, Mor مر _ Waw و , Lkhr لخر _ Eej يج , Tta لطا _ Gah قة , Alif ا _ Lttaa لطا _ Gah قة , Alif ا _ Lktaa لكتا _ Baa ب respectively. Figures 4.7(a),4.7(b),4.7(c),4.7(d), 4.7(e),4.7(f) show these examples. We actually added some special symbols into classes that we ignored classes, such as (” _ : _ . _ < _ > and ?). The reason for ignoring these classes is the great similarity (sometimes identical) between these symbol classes and other symbol classes For example, a full stop is identical in shape to the Indian numeral 0.



(a) Irregular Class Of Akthar أكثر.

(b) Irregular Class Of Moro مرو.

(c) Irregular Class Of Lkhreej لخريج

(d) Irregular Class Of Lttagaa لطاقة.

(e) Irregular Class Of Attagaa الطاقة.

(f) Irregular Class Of Alktabaa الكتاب.

Figure 4.7: The Figures a-b-c-d-e-f are Irregular Classes Of Akthar أكثر , Moro مرو , Lkhreej لخريج, Lttagaa لطاقة , Attagaa الطاقة , Alktabaa الكتاب respectively.

Each database contains Line images and paragraph image. In this thesis, we only use line images of the forms and their corresponding ground truth text. Table 4.1 shows that 346 and 74 writers were selected from the KHATT-Database for training and testing, respectively. As for IFN/ENIT, 100 and 70 writers were selected for training and testing respectively. We were limited by this number of writers due to the lengthy time consuming process that is needed to generate PAWs.

Table 4.1: Writers Number in Selected Training and Testing KHATT and IFN/ENIT Databases.

KHATT-Database	Number of Writers	IFN/ENIT-database	Number of Writers
Training	346	Training	100
Testing	74	Testing	70

The number of words ,unique words and characters for training and testing are computed for all selected line images of the two databases. From the KHATT-Database, the number of words, unique words and characters in the training data are 27570, 26696 and 128791, respectively. For the test data, there were 5605, 5441 and 26926 words , unique words and characters, respectively. On the other hand, 8124 words, 8124 unique words and 41517 characters were extracted from the line images of the training data of IFN/ENIT-Database, and 2928 words, 2928 unique words and 15135 characters were used from the line images of the testing data. Table 4.2 summarizes the statistical information of used training and testing KHATT-Database, and Table 4.3 summarizes the corresponding information for IFN/ENIT-Database.

Table 4.2: Statistical Information of data used from Training and Testing of KHATT-Database.

KHATT-Database	Words	Unique Words	Chars	Line Images	PAWs	Classes	filtering Classes	Used instances
Training	27570	26696	128791	2503	66578	3436	3023	64534
Testing	5605	5441	26926	549	13544	1449	1000	12638

Table 4.3: Statistical Information of data used from Training and Testing of IFN/ENIT-Database

IFN/ENIT-database	Words	Unique Words	Chars	Line Images	PAWs	Classes	filtering Classes	Used instances
Training	8124	8124	41517	5337	19853	679	510	19308
Testing	2928	2928	15135	1936	7731	517	381	7482

We, also, compute the number of PAWs for each PAW size. The size, here, means the number of characters in that PAW. Such information gives us a preliminary study on the nature of the division of PAWS in Database Table 4.4 shows the number of PAWs for each size in the training KHATT-Database and Table 4.5 shows the number of PAWs for each size in the testing KHATT-Database. While Table 4.6 shows the number of PAWs for each size in the training IFN/ENIT Database and Table 4.7 shows the number of PAWs for each size in the testing IFN/ENIT Database.

Table 4.4: Number of PAWs for each PAW Size in Training KHATT-Database.

PAW Size	Training	Out Training	Total
1	30832	1479	32311
2	15667	71	15738
3	10414	151	10565
4	5081	144	5225
5	1756	114	1870
6	625	49	674
7	139	23	162
8	19	9	28
9	1	2	3
10	0	0	0
11	0	1	1
12	0	1	1
Total	64534	2044	66578

Table 4.5: Number of PAWs for each PAW Size in Testing KHATT-Database.

PAW Size	Testing	Out Testing	Total
1	5989	314	6303
2	3254	20	3274
3	2096	65	2161
4	900	218	1118
5	263	171	434
6	128	82	210
7	7	33	40
8	1	2	3
Total	12638	905	13543

Table 4.6: Number of PAWs for each PAW Size in Training IFN/ENIT-Database.

PAW Size	Training	Out Training	Total
1	9739	0	9739
2	4996	137	5133
3	2748	125	2873
4	1495	154	1649
5	291	82	373
6	28	43	71
7	11	4	15
Total	19308	545	19853

Table 4.7: Number of PAWs for each PAW Size in Testing IFN/ENIT-Database.

PAW Size	Testing	Out Testing	Total
1	3838	1	3839
2	1927	44	1971
3	1043	55	1098
4	551	79	630
5	114	45	159
6	6	22	28
7	3	3	6
Total	7482	249	7731

CHAPTER 5

EXPERIMENTAL RESULTS

Different techniques have been studied on Arabic and Farsi Handwritten recognition at the PAW level over the last ten years. These techniques differ in the used databases, features and classifiers. In this chapter we will report in more details the performance of PAW image recognition on different combinations of features, classifiers, and Databases. We utilized KHATT and IFN/ENIT databases as primary databases for our training and testing. The suggested classifiers are the K-Nearest neighbor, with k=1, 3 and 5, and support vector machines (SVM). Furthermore, the suggested features are a set of four low level features CGP and Gabor Features.

The following section describes the set of experimentation carried out in this research. It is worth noting that all experimentations are done using Matlab and WEKA, as explained earlier, on 3 nodes of high performance computing services at KFUPM, two of which have 64 Gigabytes of RAM and 8 processors and the third one has 192 Gigabytes of RAM and 2 processors. With this sophisticated hardware, we were still unable to carry out some experiments, as shown below, due to ''out of memory" error. This is mainly attributed to the large size of input data.

5.1 Description of the Performed Experimentations

In our experiments, we used different combinations of features, classifiers and databases. PAW images were normalized to a fixed width and height of 128×128 when using the Gabor features, but were normalized to 128 width, preserving the aspect ratio of each PAW for the low level features. It depends on our notes of parameters implementation. Experiments included 53 of the low level features, 80 of the Gabor features and 133 of a combination of low level features and Gabor features. In addition, KHATT and IFN/ENIT databases were used in three different combinations for training and testing. The first two experiments were carried out on a data set in which both the training set and the testing set are coming from the same database, i.e. KHATT or IFN-ENIT. The third experiment was carried out with training data coming from KHATT and the testing data coming from IFN/ENIT. The motivation behind the last experiment is to study the suitability of the PAW database that was extracted from KHATT when applied to a totally different database. In the next sections we will describe experimental results for KHATT-Database and and IFN/ENIT-Database in details. In our experiments, we first extracted a set of four low-level features from the PAW images. In these experiments we used 53 features. 4 features of concentric circles centered at the centroid, 8 features are generated by dividing the image using 45 degree slices measure with respect to the centroid, 9 features of nine rectangular regions and 32 features generated by using concentric circles and four angular lines with equally-spaced angles 0,45,90 and 135 degrees (Circular Polar Grid). Regarding Gabor features, we extracted features from PAW images using the matlab code in [56]. In these experiments, we used 80 Gabor features ($8 \times 5 \times 2 = 80$) for 8 orientations, 5 scales and the mean and variance. These were experimentally found to give best results. Finally, we mixed 53 generated features of a set of four low level features with 80 of Gabor features(53+80=133).

In machine learning, different classifiers on various datasets can be used to provide a best results since no specific classifier can provide best result on different problems. The performance of the classifier can be determined by evaluating the results on the datasets. A confusing matrix clarifies the accuracy of the problem whenever the values outside the diagonal of the matrix. A confusion matrix shows information about real and predicted classifications produced by the classifier. Number of performance metrics can be derived from the confusion matrix as follows:

1. Accuracy (AC):

   The accuracy (AC) is the percentage of the total number of predictions that were correctly classified. It shows the ratio of the total number of correct classification to overall number of the classification for a single class. The common terms that are used to describe the accuracy metric are true positive (TP), true negative (TN), false negative (FN), and false positive (FP) [57]. It is defined as follows:

$$Accuracy = \frac{(TN + TP)}{(TN + TP + FN + FP)} \tag{5.1}$$

2. Recall Or TP Rate:

   It is the percentage of positive cases that were correctly classified and a measure of a classification model to select instances of a certain class from a data set.

   It is defined as:

$$Recall = TPR = \frac{(TP)}{(TP + FN)} \tag{5.2}$$

3. FP Rate:

   The false positive rate is the percentage of negative samples that were incorrectly classified as positive. It's also known as false alarm rate.

   It is defined as follows:

$$FPR = \frac{(FP)}{(TN + FP)} \tag{5.3}$$

4. Precision (P):

It is the Probability that positive cases are correctly predicted and also known as positive predictive value.

It is defined as follows:

$$Precision = \frac{(TP)}{(TP + FP)} \tag{5.4}$$

In this thesis, we have been used K-nearest neighbor(KNN) and support vector machine(SVM) as the main classifiers with different combinations of features and databases. Also, we have been used true positive rate (Recall) as main result measure of classes in our work.

5.2 Experimental Results for KHATT

This section reports the performance results achieved when both the training and testing sets were taken from KHATT. The number of used classes is 1000 classes for each part, since we ignored classes that belong to the testing data that do not have any counterpart in the training data. We, also, ignored classes that belonged to the training data that did not have samples in the test data. This resulted in a total of 59529 PAWs for training and 12638 PAWs for testing. Table 5.1 shows all used classes and PAWs for each part of KHATT-Database, and Table 5.2 shows the recognition rate of KHATT-Database of different classifiers.

Table 5.1: Number of PAWs and Classes in Training and Testing of KHATT-Database.

Database	Classes	PAWs
Training KHATT-Databse	1000	59529
Testing KHATT-Databse	1000	12638

Table 5.2: Recognition Rate of PAW Classes using KHATT-Database.

Classifier	Low level features	Gabor features	Low level + Gabor features
$NN_{k=1}$	51.67%	33.45%	54.5224%
$NN_{k=3}$	51.82%	34.94%	55.7094%
$NN_{k=5}$	53.54%	37.11%	57.3%
SVM	unable to run	unable to run	unable to run

Due to the out of memory error, SVM was not able to build the training model for the KHATT-Database. The best recognition rate achieved was 57.3% of correctly classified instances using K-nearest neighbor with k=5, using 133 features. The average accuracy of class sizes gives evidence that PAW size 1 reported high recognition rate because of their high frequency in the training model. Table 5.1 shows the true positive rate of the top 20 PAW classes using training and testing from KHATT-Database.

Table 5.3: True Positive Rate of Top 20 PAW Classes for KHATT-Database

Total	**Correct**	**True Positive Rate(Recall)**	**Class**
3	3	1	7
1	1	1	بببحه
1	1	1	سبطا
1	1	1	سعه
1	1	1	سفا
2	2	1	صله
1	1	1	لحلبد
2	2	1	لطببعه
1	1	1	هما
2552	2506	0.982	ا
69	63	0.913	م
834	757	0.908	و
51	44	0.863	هد
427	368	0.862	ر
187	158	0.845	با
326	273	0.837	،
6	5	0.833	ـ
278	227	0.817	ب
133	108	0.812	ل
5	4	0.8	فبل

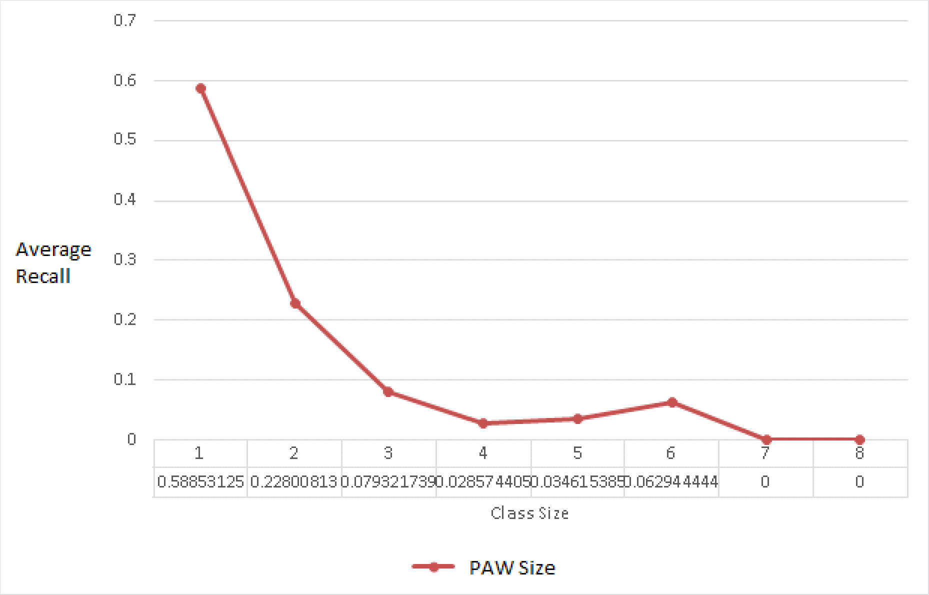


Figure 5.1: Average Recall Graphical Representation of First 6 PAW Sizes in used KHATT-Database Instances.

Obviously, PAWs that have many clear instances in the training classes model have reported good recognition rate. In KHATT-Database with k-NN k=5, information on the top performing 20 classes is shown in table 5.3.

5.3 Experimental Results for IFN/ENIT

In this section, we report the performance results for PAW recognition when the training and testing data are coming from the IFN/ENIT-Database. Similar to the justification given in the previous section, the number of used classes is 510 classes for training and 278 classes for testing, ignoring those that do not appear in either set. Hence, we got a total number of 19308 and 7341 PAWs for the training and testing sets, respectively. In addition, we ignored PAW classes in testing that had less than 3 samples in their training classes. Table 5.4 shows all used classes and PAW information for each dataset and Table 5.5 shows the recognition rates for IFN/ENIT on different classifiers.

Table 5.4: Number of PAWs and Classes in Training and Testing of IFN/ENIT-Database

Database	Classes	PAWs
Training IFN/ENIT-Database	510	19308
Testing IFN/ENIT-Database	278	7341

Table 5.5: Recognition Rate of PAW Classes using IFN/ENIT-Database.

Classifier	Low level features	Gabor features	Low level + Gabor features
$NN_{k=1}$	61.40%	38.91%	65.11 %
$NN_{k=3}$	62.04%	40.24%	66.69%
$NN_{k=5}$	63.11%	42.29%	68.16%
SVM	62.72%	47.69	73.07%

The correctly classified instances using IFN/ENIT-Databases in training and testing was 73.07% using 133 features by support vector machine. The average accuracy for the 7 sizes of PAW classes gives evidence that PAWs consisting of 1 character again reported high recognition rate due to their high number of samples in each class. Figure 5.2 shows a graphical representation of average accuracy for 7 sizes of PAW classes where both datasets are coming from IFN/ENIT.

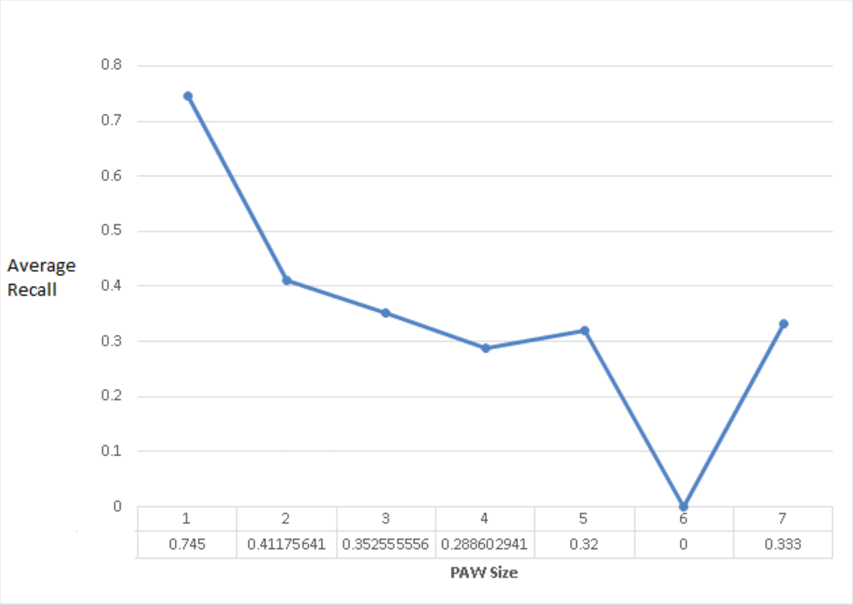


Figure 5.2: Average Recall Graphical Representation of First 7 PAW Sizes in Used IFN/ENIT-Database Instances.

The 20 top performing PAW classes are shown in table 5.6. It shows PAW sizes that have reported the best recognition rate. It shows true positive rate of top 20 PAW classes using training and testing from IFN/ENIT-Database.

Table 5.6: True Positive Rate of Top 20 PAW Classes for IFN/ENIT-Database

Total	**Correct**	**True Positive Rate(Recall)**	**Class**
1	1	1	سعد
2	2	1	صببح
1	1	1	كبد
5	5	1	لحببب
1	1	1	لحطف
2	2	1	لعلم
11	11	1	لفحص
85	84	0.99	سبد
1441	1404	0.97	ا
18	17	0.94	7
34	31	0.91	8
269	245	0.91	ه
30	27	0.9	ط
180	161	0.89	ي
36	32	0.89	بع
53	47	0.89	6
62	55	0.89	ع
17	15	0.88	حب
183	161	0.88	بو
8	7	0.88	بببص

5.4 Experimental Results for a combined KHATT and IFN/ENIT Datasets

In this section, we report results of the experiments that were carried out where the training dataset came from KHATT and the testing dataset came from IFN/ENIT. These were used to determine the feasibility of fits and value improvement of different datasets. The number of used classes after removing those not appearing in either dataset is 300 classes for each. The training dataset contained 49435 sample PAWs and the testing dataset contained 7210 PAW instances. Table 5.7 shows all used classes and PAWs for each dataset and Table 5.8 shows the recognition rates when different classifiers were used.

Table 5.7: Number of PAWs and Classes in Training KHATT-Database and Testing IFN/ENIT-Database.

Database	Classes	PAWs
Training KHATT-Database	300	49435
Testing IFN/ENIT-Database	300	7210

Table 5.8: Recognition Rate of PAW Classes using Training KHATT-Database and Testing IFN/ENIT-database.

Classifier	Low level features	Gabor features	Low level + Gabor features
$NN_{k=1}$	46.14%	30.85%	48.793%
$NN_{k=3}$	46.99%	32.23%	50.80%
$NN_{k=5}$	47.46%	33.55%	51.15%
SVM	45.60%	35.53%	53.99%

In this case, the best recognition rate achieved was 53.99% of correctly classified instances using 133 features by SVM. Figure 5.3 shows a graphical representation of average recall of First 5 PAW Sizes.

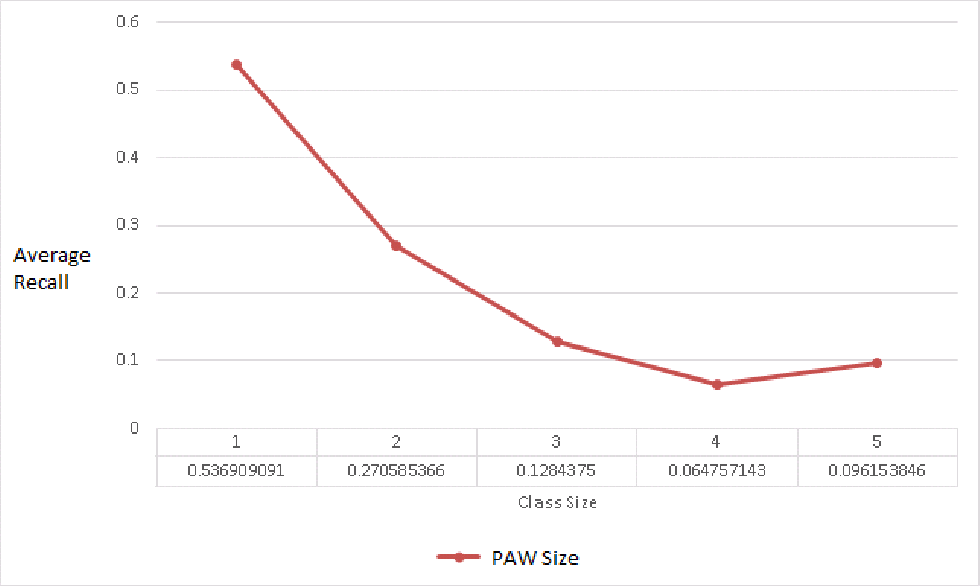


Figure 5.3: Average Recall Graphical Representation of First 5 PAW Sizes in Used training KHATT-Database and Testing IFN/ENIT-Database Instances.

Also, the best TPR rate for the top 20 performing classes are given in Table

5.9.It shows PAW sizes that have reported the best recognition rate. It shows true positive rate of top 20 PAW classes using training data from KHATT-Database and testing data from IFN/ENIT-Database.

Table 5.9: True Positive Rate of Top 20 PAW Classes for training KHATT-Database and testing IFN/ENIT-Database

Total	**Correct**	**True Positive Rate(Recall)**	**Class**
1	1	1	بببحه
1	1	1	حبا
2	2	1	حد
1	1	1	كبد
1	1	1	كبن
1	1	1	لعلا
1441	1401	0.972	ا
35	30	0.857	ببه
220	187	0.85	و
6	5	0.833	فه
5	4	0.8	علي
5	4	0.8	لحبو
68	52	0.765	به
105	80	0.762	ن
141	107	0.759	با
36	27	0.75	بع
4	3	0.75	لمد
118	88	0.746	ب
259	199	0.74	ه
539	398	0.738	ر

We evaluate our work by comparing this work to the work of [34]. This work carried out tests on two different databases, one open-vocabulary and the other one is closed-vocabulary. The number of classes is more than 1000 classes, reporting 73% of recognition rate using CPG features with Gabor features. While [34] reported 84.3% of recognition rate using GSC mixed with Gabor features on only 34 most frequent PAW selected classes.

CHAPTER 6

CONCLUSIONS

In this thesis, we investigated the use of PAWs in handwriting text recognition. We developed a PAW generation and identification system for Arabic handwritten documents. We used KHATT and IFN/ENIT databases to develop a database of PAWs and study the most frequent PAWs and divide them into different classes. Different image features and classifiers, such as the Euclidean distance and support vector machine classifiers were used to recognize PAWs and isolated characters.

6.1 Thesis Contributions

The most significant contribution of the thesis is the development of a framework to extract PAWs from Arabic handwritten text images of lines. As a result, a database of PAWs with its corresponding ground truth was developed, based on two important Arabic handwritten text databases, viz., KHATT and IFN/ENIT. We also built a clustering system for our dataset. Also, in this work, we examined different approaches for PAW Identification. We used different techniques to classify PAW images. We conducted experimentation using the generated database of PAWs and found that PAW recognition in Arabic is a promising direction of research.

6.2 Limitations of Work

There are some limitations to our work. Firstly, we have two Databases in which one of them is a proprietary databases and the other one is an open database. In the proprietary database, the number of repeated instances in each classes is large and thus have positively affected the final results of PAW identification. In the open KHATT-database, PAW classification deals with scenarios where the set of classes is not known in advance. It should be able to detect the pattern among all PAWs since no label for PAWs is given during model building. Secondly, our developed algorithm for PAW extraction cannot fully automatically succeed in extracting connected components when the text is unclear or for those who write the same text in different sizes. The framework has to manually resolve these problems.

6.3 Future Work

Our research work can be extended into four main directions in the future:

- Building a comprehensive PAW database that contains many instances of frequently appearing PAWs in text may prove to be crucial for improving PAW recognition results. This necessitates conducting a more careful study of the distribution of PAWs in Arabic, and how much of it is covered by those generated in this thesis.

- Types of features are very important in recognition problems, and different features affect results greatly. The best result of our work was using Gabor features mixed with statistical features. Therefore, amending the statistical features with Gabor features gives better recognition rates for PAWs. There is still much room to investigate other features to enhance recognition.
- Applying different techniques for segmentation of PAW images and, possibly, distributing Gabor features among different segments of PAW Images.

REFERENCES

[1] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, H. Amiri *et al.*, "Ifn/enit-database of handwritten arabic words," in *Proc. of CIFED*, vol. 2. Citeseer, 2002, pp. 127–136.

[2] S. A. Mahmoud, I. Ahmad, M. Alshayeb, W. G. Al-Khatib, M. T. Parvez, G. A. Fink, V. Märgner, and H. El Abed, "Khatt: Arabic offline handwritten text database." in *ICFHR*, 2012, pp. 449–454.

[3] B. Al-Badr and S. A. Mahmoud, "Survey and bibliography of arabic optical text recognition," *Signal processing*, vol. 41, no. 1, pp. 49–77, 1995.

[4] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 63–84, 2000.

[5] B. Bushofa and M. Spann, "Segmentation of arabic characters using their contour information," in *Digital Signal Processing Proceedings, 1997. DSP 97., 1997 13th International Conference on*, vol. 2. IEEE, 1997, pp. 683–686.

[6] T. Sari and A. Kefali, “A search engine for arabic documents,” in *Colloque International Francophone sur l’Ecrit et le Document*. Groupe de Recherche en Communication Ecrite, 2008, pp. 97–102.

[7] M. H. N. Yahia, “Content-based retrieval of arabic historical manuscripts using latent semantic indexing,” Ph.D. dissertation, KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS (SAUDI ARABIA), 2011.

[8] M. Khayyat, L. Lam, and C. Y. Suen, “Arabic handwritten word spotting using language models.” in *ICFHR*, 2012, pp. 43–48.

[9] M. T. Parvez and S. A. Mahmoud, “Offline arabic handwritten text recognition: a survey,” *ACM Computing Surveys (CSUR)*, vol. 45, no. 2, p. 23, 2013.

[10] L. Huang, F. Yin, Q.-H. Chen, and C.-L. Liu, “Keyword spotting in offline chinese handwritten documents using a statistical model,” in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 78–82.

[11] Y. Lu and C. L. Tan, “Word spotting in chinese document images without layout analysis,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3. IEEE, 2002, pp. 57–60.

[12] S. Wshah, G. Kumar, and V. Govindaraju, “Script independent word spotting in offline handwritten documents based on hidden markov models,” in

Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on. IEEE, 2012, pp. 14–19.

[13] I. S. Abuhaiba, S. A. Mahmoud, and R. J. Green, "Recognition of handwritten cursive arabic characters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 6, pp. 664–672, 1994.

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

[15] G. X. Tan, C. Viard-Gaudin, and A. C. Kot, "Automatic writer identification framework for online handwritten documents using character prototypes," *Pattern Recognition*, vol. 42, no. 12, pp. 3313–3323, 2009.

[16] U.-V. Marti and H. Bunke, "Text line segmentation and word recognition in a system for general writer independent handwriting recognition," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on.* IEEE, 2001, pp. 159–163.

[17] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *Pattern recognition (icpr), 2010 20th international conference on.* IEEE, 2010, pp. 3416–3419.

[18] T. Su, T. Zhang, and D. Guan, "Corpus-based hit-mw database for offline recognition of general-purpose chinese handwritten text," *International Jour-*

nal of Document Analysis and Recognition (IJDAR), vol. 10, no. 1, pp. 27–38, 2007.

[19] Q.-F. Wang, F. Yin, and C.-L. Liu, "Handwritten chinese text recognition by integrating multiple contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 8, pp. 1469–1481, 2012.

[20] Q.-F. Wang, E. Cambria, C.-L. Liu, and A. Hussain, "Common sense knowledge for handwritten chinese text recognition," *Cognitive Computation*, vol. 5, no. 2, pp. 234–242, 2013.

[21] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–521.

[22] T. Rath, S. Kane, A. Lehman, E. Partridge, and R. Manmatha, "Indexing for a digital library of george washingtons manuscripts: A study of word matching techniques," 2002.

[23] E. Saykol, A. K. Sinop, U. Gudukbay, O. Ulusoy, and A. E. Çetin, "Content-based retrieval of historical ottoman documents stored as textual images," *Image Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 314–325, 2004.

[24] Ö. N. Gerek, A. H. Tewfik, V. Atalay *et al.*, "Subband domain coding of binary textual images for document archiving," *Image Processing, IEEE Transactions on*, vol. 8, no. 10, pp. 1438–1446, 1999.

[25] Z. Al Aghbari and S. Brook, “Word stretching for effective segmentation and classification of historical arabic handwritten documents,” in *Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on.* IEEE, 2009, pp. 217–224.

[26] U.-V. Marti and H. Bunke, “The iam-database: an english sentence database for offline handwriting recognition,” *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[27] S. Strassel, “Linguistic resources for arabic handwriting recognition,” in *Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt.* Citeseer, 2009.

[28] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, “Database development and recognition of handwritten devanagari legal amount words,” in *Document Analysis and Recognition (ICDAR), 2011 International Conference on.* IEEE, 2011, pp. 304–308.

[29] V. Frinken, M. Baumgartner, A. Fischer, and H. Bunke, “Semi-supervised learning for cursive handwriting recognition using keyword spotting.” in *ICFHR*, 2012, pp. 49–54.

[30] A. Broumandnia, J. Shanbehzadeh, and M. Nourani, “Handwritten farsi/arabic word recognition,” in *Computer Systems and Applications, 2007. AICCSA’07. IEEE/ACS International Conference on.* IEEE, 2007, pp. 767–771.

[31] I. A. Lawal, R. E. Abdel-Aal, and S. A. Mahmoud, "Recognition of handwritten arabic (indian) numerals using freeman's chain codes and abductive network classifiers," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 1884–1887.

[32] S. Mahmoud, "Recognition of writer-independent off-line handwritten arabic (indian) numerals using hidden markov models," *Signal Processing*, vol. 88, no. 4, pp. 844–857, 2008.

[33] Z. Bahmani, F. Alamdar, R. Azmi, and S. Haratizadeh, "Off-line arabic/farsi handwritten word recognition using rbf neural network and genetic algorithm," in *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, vol. 3. IEEE, 2010, pp. 352–357.

[34] J. Chen, H. Cao, R. Prasad, A. Bhardwaj, and P. Natarajan, "Gabor features for offline arabic handwriting recognition," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010, pp. 53–58.

[35] A. Alaei, U. Pal, and P. Nagabhushan, "A comparative study of persian/arabic handwritten character recognition." in *ICFHR*, 2012, pp. 123–128.

[36] A. Acharyya, S. Rakshit, R. Sarkar, S. Basu, and M. Nasipuri, "Handwritten word recognition using mlp based classifier: A holistic approach," *International Journal of Computer Science Issues*, vol. 10, no. 2, pp. 422–427, 2013.

[37] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "Cmaterdb1: a database of unconstrained handwritten bangla and bangla–english mixed script document image," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 1, pp. 71–83, 2012.

[38] J. H. AlKhateeb, "A database for arabic handwritten character recognition," *Procedia Computer Science*, vol. 65, pp. 556–561, 2015.

[39] S. Al-Maadeed, "Text-dependent writer identification for arabic handwriting," *Journal of Electrical and Computer Engineering*, vol. 2012, p. 13, 2012.

[40] S. Mozaffari, H. El Abed, V. Märgner, K. Faez, and A. Amirshahi, "Ifn/farsi-database: a database of farsi handwritten city names," in *International Conference on Frontiers in Handwriting Recognition*, 2008.

[41] A. Mezghani, S. Kanoun, M. Khemakhem, and H. E. Abed, "A database for arabic handwritten text image recognition and writer identification," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on.* IEEE, 2012, pp. 399–402.

[42] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic handwriting recognition using baseline dependant features and hidden markov modeling," in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on.* IEEE, 2005, pp. 893–897.

[43] M. G. Kchaou, S. Kanoun, and J.-M. Ogier, "Segmentation and word spotting methods for printed and handwritten arabic texts: A comparative study." in *ICFHR*, 2012, pp. 274–279.

[44] M. A. Ali, "An efficient segmentation algorithm for arabic handwritten characters recognition system," in *Afro-European Conference for Industrial Advancement*. Springer, 2015, pp. 193–204.

[45] L. Dinges, A. Al-Hamadi, M. Elzobi, S. El-etriby, and A. Ghoneim, "Asm based synthesis of handwritten arabic text pages," *The Scientific World Journal*, vol. 2015, 2015.

[46] N. Dershowitz and A. Rosenberg, "Arabic character recognition," in *Language, Culture, Computation. Computing-Theory and Technology*. Springer, 2014, pp. 584–602.

[47] P. K. Singh, S. K. Dalal, R. Sarkar, and M. Nasipuri, "Page-level script identification from multi-script handwritten documents," in *Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on*. IEEE, 2015, pp. 1–6.

[48] P. K. Singh, A. Mondal, S. Bhowmik, R. Sarkar, and M. Nasipuri, "Word-level script identification from handwritten multi-script documents," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*. Springer, 2015, pp. 551–558.

[49] X. Wang, X. Ding, and C. Liu, "Gabor filters-based feature extraction for character recognition," *Pattern recognition*, vol. 38, no. 3, pp. 369–379, 2005.

[50] S. A. Mahmoud and W. G. Al-Khatib, "Recognition of arabic (indian) bank check digits using log-gabor filters," *Applied Intelligence*, vol. 35, no. 3, pp. 445–456, 2011.

[51] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," *Multiple Classifier Systems*, pp. 1–17, 2007.

[52] M. Law, "A Simple Introduction to Support Vector Machines," http://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf.

[53] T. Hastie, R. Tibshirani *et al.*, "Classification by pairwise coupling," *The annals of statistics*, vol. 26, no. 2, pp. 451–471, 1998.

[54] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with java implementations," 1999.

[55] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[56] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Identification using encrypted biometrics," in *Computer Analysis of Images and Patterns*. Springer, 2013, pp. 440–448.

[57] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

Vitae

- Name: Hasan Haddad Hamed Al-kaf
- Nationality: Yemeni
- Date of Birth: 02/08/1983
- Email: *alkaffhassan2010@gmail.com*
- Address Tarim-Hadramout-Yemen
- Academic Background:
  - Received Bachelor of Science (B.S.) in Computer Science from Al-ahgaff University, Yemen in 2006.
  - Joined the Information Technology Department as Lecturer assistance at SEIYUN Community College, SEIYUN, Yemen from 2006-2010.
  - Completed Master of Science (M.S.) in Computer Science from King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia in May 2015.