Sudan University of Science and Technology

College of Post Graduate Studies

Two Stages Neuro-Fuzzy System for Isolated Arabic Handwritten Character Recognition

نظام ثنائي المراحل بإستخدام هجين المنطق الغامض والشبكات العصبية للتعرف على الحروف العربية المعزولة المكتوبة بخط اليد

A Dissertation Submitted to the College of Graduate Studies,
Faculty of Computer Science and information Technology,
Sudan University of Science and Technology
In Partial Fulfillment of the Requirements for the degree of

DOCTOR OF PHILOSOPHY

Major Subject: Pattern Recognition

by

Omar Balola Ali Ahmed

Supervisor:

Adnan Shaout, Professor

Desember 2016

Examiners Approval

This is to certify that the dissertation prepared

By: Omar Balola Ali Ahmed

Entitled: Two Stages Neuro-Fuzzy for Isolated Arabic Handwritten Character Recognition and submitted in partial fulfillment of the requirements for the degree of doctor of philosophy of Computer Science.

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

Signed by the final examining committee:

Signature Date of Signature

(dd/mm/yyyy)

Examiner 1 Signature

Examiner 2 Signature

Examiner 1 Signature

Examiner 2 Signature

Declaration of the Status of Thesis (by student)

I, the signing here-under, declare that the contents of this dissertation represent my own work, and that the dissertation has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions, which is an original intellectual work.

Candidate’s name: Omar Balola Ali Ahmed
Candidate’s signature:
Date:

Declaration of the Status of Thesis (by Supervisor)

I, the signing here-under, declare that I'm the supervisor of the sole author of the Ph.D. dissertation entitled:
Two Stages Neuro-Fuzzy for Isolated Arabic Handwritten Character Recognition

Supervisor's name: Prof. Adnan Shaout
Supervisor's signature:
Date:

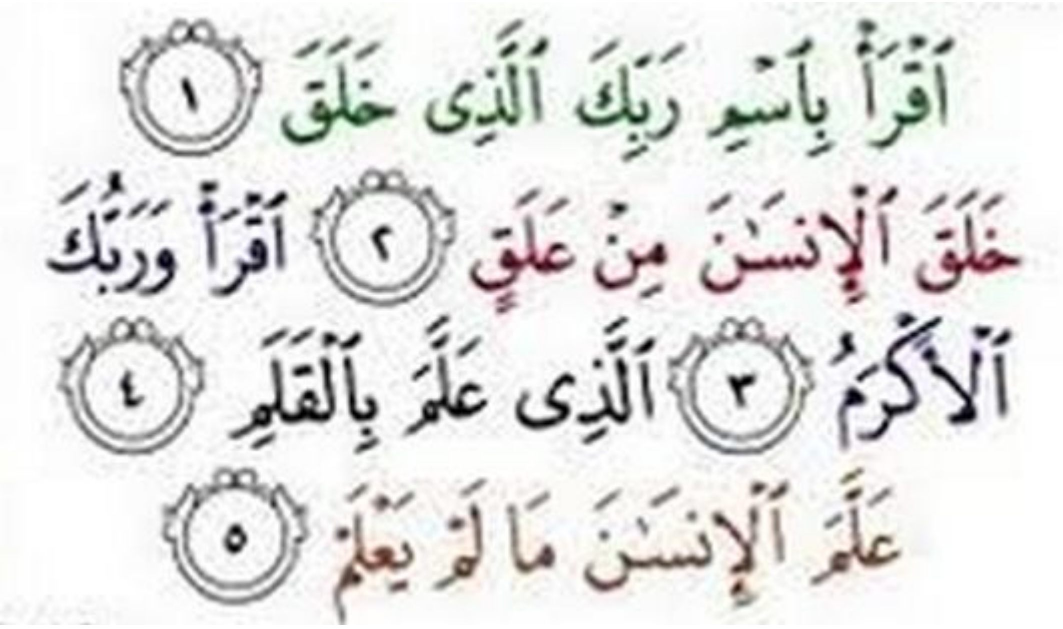
Assigning the copy-right to CGS

I, the signing here-under, declare that I'm the sole author of the Ph.D. dissertation entitled:
Two Stages Neuro-Fuzzy for Isolated Arabic Handwritten Character Recognition
This is an original intellectual work. Willingly, I assign the copy-right of this work to the College of Graduate Studies (CGS), Sudan University of Science and Technology (SUST).

Accordingly, SUST has all the rights to publish this work for scientific purposes.

Candidate's name: Omar Balola Ali Ahmed
Candidate's signature:
Date:

بسم الله الرحمن الرحيم



ٱقْرَأْ بِٱسْمِ رَبِّكَ ٱلَّذِى خَلَقَ ﴿١﴾
خَلَقَ ٱلْإِنسَٰنَ مِنْ عَلَقٍ ﴿٢﴾ ٱقْرَأْ وَرَبُّكَ
ٱلْأَكْرَمُ ﴿٣﴾ ٱلَّذِى عَلَّمَ بِٱلْقَلَمِ ﴿٤﴾
عَلَّمَ ٱلْإِنسَٰنَ مَا لَمْ يَعْلَمْ ﴿٥﴾

Dedication

*To my mother, t*o my wife, who has given me all the support for my success.

To my sons and daughters, my teachers, and all my friends, support and encouragement

ACKNOWLEDGEMENT

The thanks first and foremost to God Almighty to reconcile me to complete this thesis And I then like to extend my deep thanks to Professor Adnan Shaout for giving me his golden time and I thank him for his guidance and patience throughout the process of researching and writing this dissertation For his follow-up to our progress and assist us with references, and his continued advised for me for the diligence and perseverance.

I would like to thank the gallant commander Professor Dr. Izzeldin Osman, who sponsors and pursue this PhD program.

I would like to thank Dr. Mohammed Elhafiz who encourages me to start my PhD program with this successful and advanced program.

I express my great gratitude and emeritus at the Faculty of Computer Science and Information Technology and the wise leadership and cooperating with us in everything we need. Dr. Talat the dean of faculty, Dr. Intsar the coordinate for third patch and the technical supporter Miss. Sara which stayed up nights to help us.

I would also like to thank my PhD colleagues in the third patch inside and outside Sudan.

Last but not least, thank my wife for the mental support and great Patience on my frequent absence.

ABSTRACT

In this thesis a two phase method for Isolated Arabic Handwritten Character Recognition (IAHCR) system has been presented. The objective of the proposed method is to achieve the best possible recognition accuracy. The new method combines two stages based on two classifiers, a public and a private, according to the similar features among characters. In the first stage, a public classifier to deal with all character groups has been built, where each group contains characters with overlapped features. The public classifier classifies the characters in the data set to a specified group. In the second stage, a private classifier for each group to recognize and classify the characters within a group has been created. To determine the similarities between the Arabic characters, a neural network algorithm using back propagation algorithm with two experiments with different data set sizes has been applied. The characters with the same structure are then grouped in one class and a new size of data set which consists of fifteen groups (classes) is formed for all the Arabic characters.

The Sudan University for Sciences and Technology Arabic Recognition Group (SUST-ARG) data set was used to classify the Arabic handwritten character groups.

In this thesis, two types of statistical features have been employed. The first type of features, called CCOB, which includes the center mass (xt, yt) of the character image, count the crosshair, the outliers (Right, Left, Top and Down) and the black ink histograms. The second set of features is based on extract eigenvectors and eigenvalues of the shifted mean of resize cropped images. The two set of features were then combined and reduced to four features using the principal component analysis PCA technique.

The Adaptive Neural network Fuzzy Inference System (ANFIS) classifier was used at all levels of the character recognition stages with different learning algorithms. For the first level, a general classifier for 34 classes was used. For the second level, a group classifier for 15 groups was used and a character classifier for separated character was also used.

Experimental results based on data set of 6800 images using Arabic Handwritten characters have proved the efficiency of the new proposed recognition system.

Different experiments have been conducted with different set of features for two stages of classification and obtained the highest recognition rate. In the first stage, the recognition rates were 96.1, 96.2 and 97.15 for the first set of features, second set of features and combined set of features, respectively. In the second stage, the recognition rates were 99.30, 99.46 and 99.34 for the first set of features, second set of features and combined set of features, respectively.

The system has achieved the highest recognition rate of 99.46% for the tested data set using the proposed two stage recognition system.

The learning process for a large training data set needs more time and requires large memory.

المستخلص

يقدم هذا البحث طريقة ثنائية المراحل للتعرف على الحروف العربية المعزولة المكتوبة بخط اليد (IAHCR). والهدف من هذه الطريقة المقترحة لتحقيق أفضل دقة ممكنة للتعرف. وتقوم هذه الطريقة الجديدة بالدمج بين مستويين استنادا الى مصنفين، مصنف عام وآخر خاص، وتم ذلك وفقا للسمات والخصائص المتشابهة بين الحروف العربية. في المستوى الأول، تم تصميم المصنف العام الذي يتعامل مع مجموعات الحروف، حيث تحتوي كل مجموعة على الحروف المتشابهة في السمات. يقوم المصنف العام بتصنيف الحروف من مجموعة بيانات الحروف العربية إلى المجموعات المحددة. في المستوى الثاني، تم تصميم مصنف خاص لكل مجموعة ليقوم بالتعرف وتصنيف الحروف ضمن المجموعة. لتحديد أوجه التشابه بين الحروف العربية تم تطبيق خوارزمية الشبكة العصبية باستخدام نموذج الانتشار الخلفي مع نوعين من التجارب والتي تم فيها استخدام احجام مختلفة من مجموعة البيانات. ثم تم تجميع الحروف المتاشبهة في الشكل في صنف واحد و تم تشكيل حجم جديد من مجموعة البيانات يتكون من خمسة عشر مجموعة (الصنف) لجميع حروف اللغة العربية. وقد استخدم لتدريب هذه المصنفات مجموعة الحروف العربية المنفصلة المكتوبة بخط التي تم إنشاءها من مجموعة تكنلوجيا اللغة العربية العربية بجامعة السودان للعلوم والتكنولوجيا.

في هذه الأطروحة، تم استخدام نوعين من تقنيات السمات الإحصائية. ، النوع الأول من هذه التقنيات، ويسمى CCOB ، وتقوم هذه التقنية بحساب كتلة المركز (Center of mass) لصورة الحرف، حساب نقاط الانتقال من الخلفية الى الواجهة من خلال الخطوط الافقية والعمودية (crosshair)، و ايجاد مسافة القيم المتطرفة(Outliers) (يمين، يسار، أعلى وأسفل) ورسوم بيانية الحبر الأسود(Black in histogram). وتستند المجموعة الثانية من السمات على استخراج المتجهات الذاتية (Eigenvectors)والقيم الذاتية(Eigenvalues) للمتوسط التي تحولت بعد تغيير حجم الصورة التي تم استبعاد الاجزاء الفارغة منها. ثم تم دمج مجموعة السمات السابقة وتخفيض حجمها إلى أربع خصائص أساسية باستخدام تقنية تحليل المكون الرئيسي PCA .

تم استخدام مصنف هجين من نظام المنطق الغامض والشبكات العصبية (ANFIS) وتطبيقه على جميع مستويات مراحل التعرف على الحروف مع خوارزميات تعلم مختلفة. للمستوى الأول، تم استخدام المصنف العام لعدد 34 صنف يحتوي كل صنف على مجموعة من صور لحرف واحد. بالنسبة للمستوى الثاني، تم استخدام مصنف المجموعة لعدد 15 مجموعة، واخيرا تم استخدام مصنف منفصل لكل حرف على حدى.

وقد أثبتت نتائج التجارب التي اجريت على عدد 6800 صوة من مجموعة بيانات الحروف العربية المكتوبة بخط اليد التي تم استخدامها كفاءة نظام التعرف الجديد الذي تم تصميمه.

أجريت تجارب مختلفة مع مجموعة مختلفة من السمات باستخدام مصنف في مرحلتين تم الحصول على أعلى معدل لدقة التعرف. في المرحلة الأولى كانت معدلات التعرف 96.1%، 96.2% و 97.15% لخصائص المجموعة الأولى، مجموعة السمات الثانية ومجموعة الهجين، على التوالي. في المرحلة الثانية كانت معدلات دقة التعرف 99.30، 99.46 و 99.34. حقق النظام أعلى معدل دقة للتعرف 99.46% لمجموعة بيانات الاختبار باستخدام نموذج المرحلتين المقترح.

TABLE OF CONTENTS

LIST OF TABLES1

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANFIS	Artificial Neural Network Fuzzy Inference System
ANNs	Artificial Neural Networks
BPNN	Back Propagation Neural Network
CCOB	Center mass Cross hair Outliers Black ink histograms
FE	Features Extraction
FL	Fuzzy Logic
HACR	Handwritten Arabic Character Recognition
HLA	Hybrid Learning Algorithm
HRR	Highest Recognition Rate
IAHCR	Isolated Arabic Handwritten Characters Recognition
OCR	Optical Character Recognition
PCA	Principle Component Analysis
PR	Pattern Recognition
SUST-ARG	Sudan University for Sciences and Technology Arabic Research Group

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

1.1 Overview

The main goal of handwriting recognition system is to transform handwritten text input from specific source, such as paper documents, photographs, touch-screens and other devices, into printed output form. From digital image format, represented inside the electronic machine to coded character format documents in order to make it readable and editable using word processing application systems. Handwritten recognition system represents an attempt to simulate the human ability in reading [1].

Handwritten Arabic letter recognition is a big problem facing technical, heritage and manuscript scientists because of the high similarity among characters in the form of letters and body structure.

Arabic language has features and characteristics which are more complex than the Latin language. The Arabic letters have more than one shape; one shape for the letter at the beginning of a word, in the middle, at the end, and isolated shape. Arabic language has many handwriting styles. An accurate feature set should be extracted from a wide sample of handwriting styles to be used for training and to have an accurate result [2].

The handwriting recognition systems can be classified into two main groups, off-line and on-line recognition, according to the format of handwriting inputs. On-line character recognition, the computer recognizes the symbols as they are drawn. The second type, the off-line character recognition, refers to the process of recognizing characters that have been scanned from a surface. Many applications require off-line handwritten character capabilities such as bank processing, mail sorting, document archiving, commercial form-reading, office automation, etc. [3]. Therefore, offline handwritten recognition is an application of pattern recognition.

Fuzzy logic was first introduced by Zadeh [4]. It was developed for solving decision making problems through the use of "IF-THEN" rules. It was used later to model uncertainty and imprecision in data management. Fuzzy logic is an easy way to reach definitive conclusions based on vague, ambiguous, imprecise and noise information [5, 6]. More detail of fuzzy logic with character recognition is described in chapter 3.

Neural Network approach is an important method in the field of handwritten character recognition. The modern theories of neural network processing and digital computers appearance was during the late 1940s. Neural Network structure is a collection of parallel processors connected together in the form of a directed graph [7]. There are many structures of artificial neural networks including, Perceptron, Adaline, Madaline, Kohonen, Back Propagation and many others. Back Propagation is the most commonly used with pattern recognition. Further detail of neural network application with character recognition area is briefly described in chapter 3.

Hybrid approaches could be considered as one of the main contribution to soft computing with neuro-fuzzy systems being the first and probably the most successful hybrid approach till now. Neuro-fuzzy systems incorporate the elements from fuzzy logic and neural networks. The idea of hybridization originates from the following two observations:

- Fuzzy systems are not capable of performing parallel computation, whereas these characteristics are clearly attributed to neural networks.
- Neural networks lack flexibility and human interaction which lies at the core of fuzzy logic.

In this dissertation, a recognition system with two stages for Isolated Arabic Handwritten Characters Recognition (IAHCR) using an Artificial Neural Network Fuzzy Inference System (ANFIS) will be presented.

1.2 Problem Statement and its Significance

In the Arab and Muslim world, manuscripts and ancient writings need to be saved, processed, edited, coordinated and presented. There are many researchers who spend a long time typing reports that were handwritten in their workplaces. Also, many citizen official papers that are handwritten would need to type to be saved and edited for future use.

This research focuses on the problem of producing an intelligent system that is capable of translating Arabic handwritten letters to print digitized letters to be saved as text.

The shapes of Arabic handwritten characters differ among writers, but the geometrical features are always the same. An important difference of Arabic handwritten characters from Latin ones is the existence of dots. Dots differentiate among characters with the same geometry.

There are strong similarities between the Arabic letters in shape, but there are some differentiators such as having the dot or not in (ع، غ) or in the number of dots such as (ت، ث) and also the position of the dot being below or above the character such as (ب، ن).

1.3 Questions

What are the proper techniques that will be used in the preprocessing stage?

What is the technique that will be used to extract the features from a character (letter) and what is the suitability of the extracted features? (a new performance evaluation measure will be generated to measure the suitability of features) What is the appropriate method for character recognition process and what is their suitability? (a new performance evaluation measure will be generated to measure the suitability of the recognition process)

What is the appropriate technology to be used to implement the recognition system?

1.4 Hypothesis

The hypothesis of this research is to produce an Arabic handwritten letter recognition software system using a suitable pattern recognition approach. The software also should give a high rate of accuracy. The approach that will be used in this research is the neuro -fuzzy approach.

This system allows the user to enter handwritten Arabic letters and output the printed letters.

1.5 Research Objectives

Researchers are still active in solving the problem of automatic reading of handwritten letters and invent ideas to improve the process. The significant benefits of the new process should be to shorten the time and reduce effort for humans to use it. The following are the main objectives for the research in this dissertation:

- To find an appropriate method to design a system to recognize Arabic handwritten characters with good rate of recognition and fast speed.
- To design software product that can recognize Arabic Characters by analyzing its shape and comparing its features against a set of rules that distinguishes each character.
- To convert the handwritten Arabic text file into one or more of the output formats offered by software including WORD, RTF and PDF.
- To enable automatic reading of ancient Arabic manuscripts.
- To help in the process of handwritten Arabic text file recognition. Handwritten Arabic text file comes in as input. It then gets segmented into handwritten character using our proposed ANFIS system which outputs the printed characters. A collection step (algorithm) will combine the characters at hand to one or more of the output formats offered by software including WORD, RTF and PDF.

- A new ANFIS system will be developed which resolve the problem of the similarities among the Arabic characters by applying a new idea using a two-stage character recognition system.
- To be a pioneer among researchers to devise new and effective method to solve the problem of similarities in the structural of the handwritten Arabic letters.

1.6 Scope of the Research

The research in Handwritten Arabic Character Recognition (HACR) usually consists of several stages such as presenting images of characters, scan these images, preprocessing and extracting features. The last stage includes training, classification and output results.

The research problem will include theory and application, where we will come up with theoretical results regarding Arabic handwritten recognition.

1.7 Research Contributions

The main contributions of this dissertation can be summarized as follows:

- Implement a hybrid Neuro-fuzzy algorithm to recognize Isolated Arabic Handwritten Letters (HAL).
- Use many image preprocessing and types of feature extraction techniques and use a large data set of (HAL) images written by 141 writers all of them repeat the letter ten times.
- Propose method that can divided the problem of HAC to sub-problems.

1.8 Thesis Layout

Chapter one is an introduction to the research topic. Chapter two explains the following concepts: pattern recognition, OCR structure, handwriting Arabic character recognition, digital image preprocessing methods, feature extraction techniques, covers the concepts of neural network, fuzzy logic and neuro-fuzzy. Chapter three presents literature review. Chapter four explains the research methodology. Chapter five presents the SUST-ADG data base and the implementation process of neural network system for determining the overlap among Arabic characters. Chapter six presents the experimental results of using the ANFIS classifiers and discusses the result. Chapter seven provides the conclusion, recommendation and further work.

CHAPTER II

BACKGROUND

This chapter introduces pattern recognition, character recognition, and Arabic handwritten character recognition. It introduces the main stages involved in developing any OCR application. The main focus of this research is Arabic Handwritten OCR and present character recognition processing stages for it, preprocessing, features extraction and recognition methods.

2.1 Pattern Recognition (PR)

In this section, presented definition of PR, applications of PR and PR methods.

2.1.1 Pattern Recognition definitions

Pattern Recognition is the research area that studies the operation and design of systems that recognize patterns in data.

Pattern Recognition is one of a most important mental ability, which is characterized by the humans. Humans can extract information from the surrounding and humans also have the ability to keep this information and use it when requested. At this time, the digital computer technology has been largely used and developed to simulate this human ability, the simulation of this human ability with automated machines is becoming more realistic [8].

Fukunaga [9] defined pattern recognition as” A problem of estimating density functions in a high-dimensional space and dividing the space into the regions of categories of classes.”

Gonzalez, Thomas [10] defined pattern recognition as a classification of input data via extracting important features from a lot of noisy data. Robert P.W. Duin [11] described the aim of pattern recognition is to design machines to solve the gap between application and theory.

2.1.2 Pattern Recognition Applications

There are many pattern recognition applications. The following are some:

- Optical character recognition will be detailed in Section 2.2
- Biomedical (Neuroscience, ECG monitoring, drug development, DNA sequences).
- Speech recognition.
- Industrial inspection.
- Biometric (face recognition, fingerprint, iris recognition).
- Military applications (Automated Target Recognition).

2.1.3 Methods of Pattern Recognition

Pattern recognition uses many methods in the development of numerous applications in different fields. The workability of these methods is intelligent emulation [12]. The following are the main methods that are used in pattern recognition:

2.1.3.1 Statistical Methods

Statistical Methods are mathematical formulas, models, and techniques that are used in statistical analysis of classification of data, which aims to minimize the loss of classification with given loss matrix and estimated probabilities [13].

2.1.3.2 Structural Methods

Structural methods concerned with the structural description of the pattern (string, tree, or graph) of flexible size. Structural representation recorded sequence stroke or topographic form of the character style, and hence resembles well to the mechanism of human perception. The structural convergence procedure not only provide the overall similarities, but also explains the structure of the input patterns and refers to the similarities between the components [14].

2.1.3.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are enormous concentrations of overlapping nodes that work together in harmony and cooperation. ANNs were initially studied with the hope of making intelligent perception and cognition machines by simulating the physical structure of human brains. The principles and algorithms of ANNs have found numerous applications in diverse fields including pattern recognition and signal processing [10]. ANNs will be detailed in Section 2.4.

2.1.3.4 Fuzzy Logic (FL)

The thinking process of human being is often fuzzy and uncertain, and the languages that humans use are also often fuzzy. In reality, the human does not always give complete answers or classifications, so the fuzzy set theory comes into being [14].

Fuzzy logic is an approach of computing which is based on "degree of truth" rather than the usual "true or false" (1 or 0) results. The idea of fuzzy logic was first introduced by Dr. Lotfi Zadeh who was working on the problem of computer understanding of natural languages. Natural languages (like most other activities in life and indeed the universe) are not easily translated into the absolute terms of 0 and 1 [15]. FL will be detailed in Section 2.5

2.2 Character Recognition (CR)

In this section the overview of character recognition, CR history, OCR application and OCR family will be presented.

2.2.1 Overview

Character Recognition or Optical Character Optical Character Recognition (OCR) is a technology that enabled to convert the character images of machine printed or handwritten text (numerals, letters, and symbols) into a computer process able format (such as ASCII, word and PDF). This is the technology long used by libraries and government agencies to make lengthy books, reports and manuscripts quickly available electronically. Advances in OCR technology have spurred its increasing use by enterprises. Before the start of using the OCR, documents were scanned and only stored as images on the media with a large space and could not be copied nor modified. OCR is an important research area in pattern recognition. The objective of an OCR system is to recognize alphabetic letters, numbers, or other characters, which are in the form of digital images, without any human intervention [16].

2.2.2 Character Recognition History

According to Arica and Yarman-Vural in their review of character recognition (CR), the CR systems have evolved in three stages [17]. The first stage is in the period of 1900-1980. The beginning of OCR was considered for the aim of producing a device to help the blind people in reading. In the second period of development in the era of the 1980s to 1990s, the explosion of information technology has helped a rapid growth in the area of OCR. The CR research was important only for techniques that recognizes the character shape. The real progress in OCR systems has been achieved on the 1990s. In the beginning of this period, the artificial intelligence is combined with some concepts, such as image processing and pattern recognition techniques. Complex algorithms for character recognition systems were developed. There is, however, still a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading, especially for unconstrained on-line and off-line handwriting [18].

2.2.3 OCR Applications

There is a wide variety of OCR software available on the market with different prices and degrees of accuracy. Some of these software applications are commonly used with the English language and other used with the Arabic language. There are various types of OCR software such as the following: Simple Software 2012, ABBYY 2012, OmniPage Professional 18 (ScanStore 2011), Readiris Pro 12 (IRIS 2011), (World 2010; Network 2012), NovoVerus from Novo Dynamics (Dynamics 2012) and OCR from Sakhr software (Software 2011) [19].

2.2.4 Different Families of Character Recognition (CR)

There are two types of character recognition as shown in figure 2.1. The first type is the on-line CR, in which the computer recognizes the symbols as they are drawn. The second type is the off-line CR, which refers to the process of recognizing characters that have been scanned from a surface (such as a sheet of paper) and are stored digitally in gray scale format [20]. This last has two types, Magnetic Character Recognition (MCR) and optical character recognition.

Optical Character Recognition (OCR) this type of characters acquiring by a scanner or a camera. The characters are in the form of pixelized images, and can be either printed or handwritten, of any size, shape, or orientation. The OCR can be subdivided into handwritten character recognition and printed character recognition. Handwritten Character Recognition is more complicated to implement than printed character recognition because the differences in the writing of the letter from one person to another. In printed character recognition, the images of Characters can be in specific fonts like Times New Roman, Arial, Courier, etc. [20]. OCR systems are being developed for all world languages and Arabic language is one of them. Arabic character recognition will be presented in the next section.

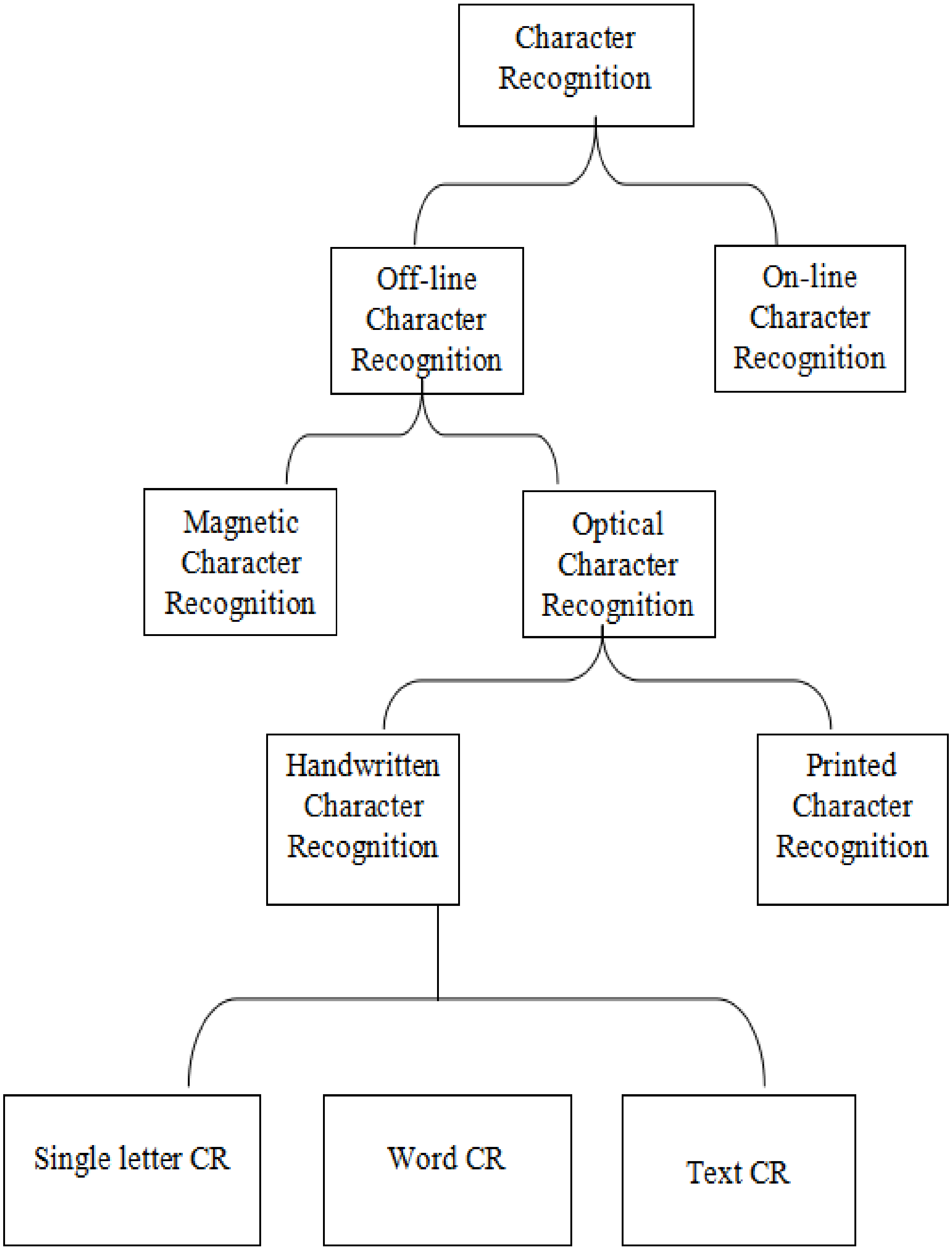

Character Recognition
Off-line Character Recognition
On-line Character Recognition
Magnetic Character Recognition
Optical Character Recognition
Handwritten Character Recognition
Printed Character Recognition
Single letter CR
Word CR
Text CR

Figure 2. 1: The different families of character recognition.

2.3 Arabic Characters Recognition (ACR)

In this section the main types of written text, Arabic aphanitic and character characteristics will be presented.

2.3.1 Overview

Arabic is a language spoken by Arabs peoples in over 22 countries, and roughly associated with the geographic region of the Middle East and North Africa, but is also spoken as a second language by several Asian countries in which Islam is the principle religion (e.g. Indonesia). However, non-Semitic languages such as Farsi, Urdu, Malay, and some West African languages such as Hausa have adopted the Arabic alphabet for writing [21]. Arabic, one of the six United Nations official languages, is the mother tongue of more than 300 million people. During the 7th century new Arabic letters were created by adding dots to existing letters to avoid ambiguities by Alhajaj Bin Yousif during Bani Umiha ruling [22].

There are two Main Types of Written Arabic Text. The first is Classical Arabic (CA), also known as Quranic Arabic, which is the form of the Arabic language used in literary texts from Umayyad and Abbasid times (7th to 9th centuries). The ancient Arab tribes are the basis of this language dialect. [22]. The second is Modern Standard Arabic (MSA), which is the Academic Language and the language of the vast majority of written material and formal TV shows, lectures, etc. [22]. Arabic handwritten character recognition system will be detailed in Section 2.4.

Due to the cursive nature of the script, there are several characteristics that make recognition of Arabic distinct from the recognition of many world languages such as Latin scripts or Chinese as shown in figure 2.2.

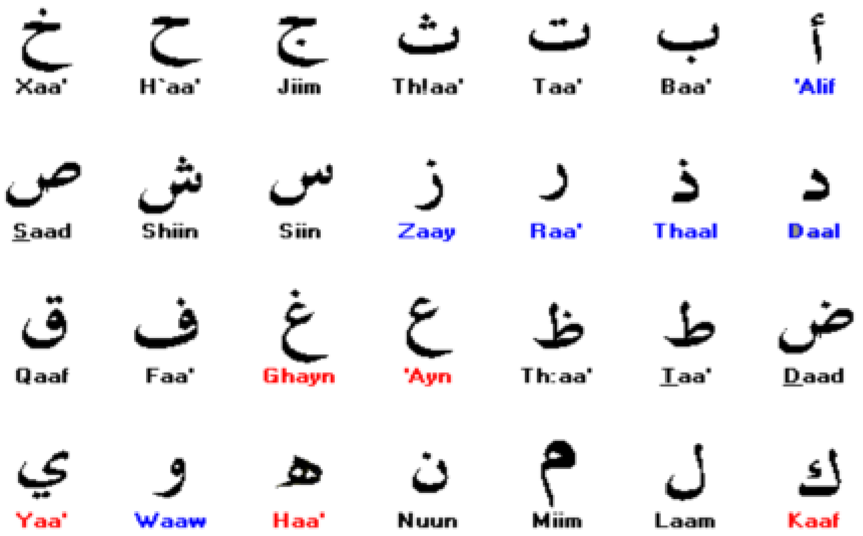


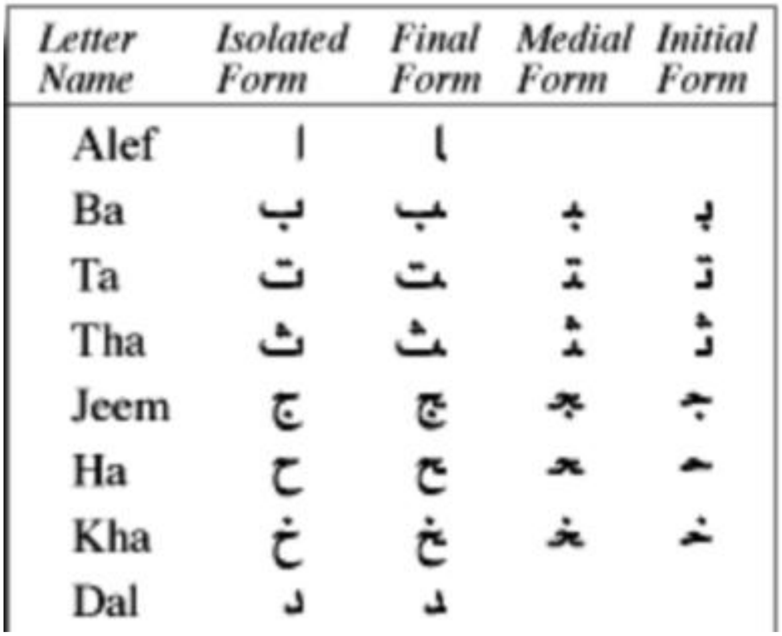
Figure 2. 2: Letters of the Isolated Arabic Alphabet.

2.3.3 Arabic Alphabet

Arabic language has basic 28 letters in the alphabet. It is based on 18 distinct shapes that vary according to their connection to preceding or following letters. Using a combination of dots and symbols above and below these shapes, the full complement of 28 consonants can be constructed. The Arabic character set is composed of 28 basic characters. Fifteen of them have dots and 13 are without dots. Dots above and below the characters, play a major role in distinguishing some characters that differ only by the number or location of dots [23].

Arabic is a script language. There are no big letters and some letters are isolated in the word (letters in Figure 2.3).

Arabic is written from right to left. Arabic has four forms for each letter depending on the position of the letter in each word. These are beginning, middle, end and isolated as shown in Figure 2.3 [23]. Arabic contains dots and other small marks that can change the meaning of a word.



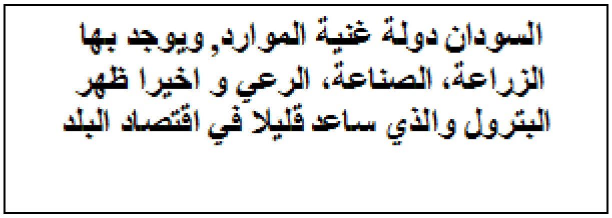
Letter Name	Isolated Form	Final Form	Medial Form	Initial Form
Alef	ا	ـا		
Ba	ب	ـب	ـبـ	بـ
Ta	ت	ـت	ـتـ	تـ
Tha	ث	ـث	ـثـ	ثـ
Jeem	ج	ـج	ـجـ	جـ
Ha	ح	ـح	ـحـ	حـ
Kha	خ	ـخ	ـخـ	خـ
Dal	د	ـد		

Figure 2. 3: Samples of Various Arabic Letter Forms.

2.3.4 Characteristics of Arabic Characters

This section presents a list of characteristics of the Arabic characters with figures to illustrate the concepts. The characteristics are as follows:

1. Arabic language is always written from right to left as shown figure 2.4. Arabic has 28 basic characters as shown in Table 2.1. No upper or lower cases exist in Arabic.



السودان دولة غنية الموارد, ويوجد بها
الزراعة، الصناعة، الرعي و اخيرا ظهر
البترول والذي ساعد قليلا في اقتصاد البلد

Figure 2. 4: A sample of printed Arabic text.

2. The shape of the character varies according to its position in the word as can be seen in table 2.1. Each character has either two or four different forms.

3. Clearly, the six characters (و, ز, ر, ذ, د, ا), if appeared in a word, will cause the word to be divided into blocks of connected components called sub words.

4. Character width differs from one character to another.

5. Fifteen characters have dots associated with the character. They can be above or below the primary part (refer to table 2.1).

6. Some characters share the same primary part [24] and distinguished from each other by the secondary part (the dots), see table 2.1.

7. Some characters contain closed loop (refer to table 2.1).

8. Hamza (ء) zigzag shape, is not really a letter, it is a complementary shape.

9. There are only three characters that represent vowels, و, ا or ي. However, there are other shorter vowels represented by diacritics in the form of over-scores or underscores (Table 2.1).

10. Dots may appear as two separated dots, touched dots, hat or as a stroke.

2.4 Arabic Handwritten Character Recognition (AHCR)

In this section described introduction, applications of Arabic handwritten character recognition systems, AHCR difficulties and system structure of AHCR.

2.4.1 Introduction

Handwritten Arabic Character Recognition (HACR) is the system that attempts to recognize a text that has been written by a person (not a machine).

Handwriting recognition attracted the attention of researchers since the origin of computers [25].

There are three types of Arabic language recognition as shown in figure 2.5. The first is single letter recognition which consists of beginning, middle, end and isolated. The second is word recognition and the last is text recognition.

Table 2. 1: Illustrates position of Arabic letters.

Character Name		Isolated	Begin	Middle	End	Others character shapes
Alef	الف	ا	ا	ـا	ـا	أ إ
Ba'	باء	ب	بـ	ـبـ	ـب	ئ
Ta'	تاء	ت	تـ	ـتـ	ـت	ؤ ء
Tha'	ثاء	ث	ثـ	ـثـ	ـث	لا
Jeem	جيم	ج	جـ	ـجـ	ـج	
H'a'	حاء	ح	حـ	ـحـ	ـح	
Kha'	خاء	خ	خـ	ـخـ	ـخ	
Dal	دال	د	د	ـد	ـد	
Thal	ذال	ذ	ذ	ـذ	ـذ	
Raa'	راء	ر	ر	ـر	ـر	
Zain	زين	ز	ز	ـز	ـز	
Seen	سين	س	سـ	ـسـ	ـس	
Sheen	شين	ش	شـ	ـشـ	ـش	
Sad	صاد	ص	صـ	ـصـ	ـص	
Dad	ضاد	ض	ضـ	ـضـ	ـض	
Taa	طاء	ط	طـ	ـطـ	ـط	
Zain	ظين	ظ	ظـ	ـظـ	ـظ	
Ain	عين	ع	عـ	ـعـ	ـع	
Gain	غين	غ	غـ	ـغـ	ـغ	
Fa	فاء	ف	فـ	ـفـ	ـف	
Gaf	قاف	ق	قـ	ـقـ	ـق	
Kaf	كاف	ك	كـ	ـكـ	ـك	
Lam	لام	ل	لـ	ـلـ	ـل	
Mim	ميم	م	مـ	ـمـ	ـم	
Non	نون	ن	نـ	ـنـ	ـن	
Ha	هاء	هـ	هـ	ـهـ	ـه	
Waw	واو	و	و	ـو	ـو	
ya	ياء	ي	يـ	ـيـ	ـي	

2.4.2 Application of Offline Handwritten Recognition

Some of the more important applications of offline handwritten recognition are discussed in the following section.

2.4.2.1 Check Reading

Offline handwritten recognition is basically used for check reading in banks. Check reading is very important for commercial application of offline handwritten recognition. Handwritten recognition system plays very important role in banks for signature verification and for recognizing the check amount filled by user [26].

2.4.2.2 Postcode Recognition

Handwritten recognition system can be used for reading the handwritten postal address on letters. Offline handwritten recognition system used for recognizing handwritten digits of a postcode. HCR can read this code and can sort mail automatically.

2.4.2.3 Form Processing

HCR can also be used for form processing. Forms are normally used for collecting public information. Replies of public information can be handwritten in the space provided [26].

2.4.2.4 Signature Verification

HCR used to recognize a person by his/her signature verification. Signature identification is the specific field of handwritten recognition, which can be used to identify a person by his/her handwriting since handwriting may vary from person to person [26].

2.4.2.5 The Media Mean

It is used in the field of media, especially in newspapers and magazines, to print columns, newspaper articles and the investigations that are written by hand by the journalists when they are in prompt conferences or field investigations.

2.4.3 Arabic Handwritten Character Difficulties

The following are difficulties that Arabic handwritten recognition faces:

1. Different writers or even the same writer under different conditions will write some Arabic characters in completely different ways.

2. Characters are cursive and not separated, as in the case of Latin script. Hence, recognition normally requires a sophisticated character segmentation algorithm.

3. Characters change shape depending on their location in the word. The distinction between isolated characters is lost when they appear in the middle of a word.

4. Sometimes Arab writers ignore the space between words when the word ends with one of the following six letters: (ا د ذ ر ز و). This is known as the connected words problem [27]. For example, (ألاستاذاحمد) should be (الاستاذ احمد). On the other hand they sometimes separate a single word into two words when the word is long or is pronounced in two parts; (سبع مائة) is actually (سبعمائة). The Arabic words need to be rewritten since they are not clear!

5. Repeated characters are sometimes used, even if this break Arabic word rules.

This is especially common in online "chat" sites; for example (مووووت) is actually (موت).

6. There are two ending letters (ي and ى) which sometimes indicate the same meaning but are different characters. For example (السوداني) and (السودانى), have the same meaning, the first is correct but the second form is less used.

7. The letter Alef (ا) has different shapes (أ إ آ). Some Arab writers do not distinguish among these differences, which would lead to little change in pronunciation and sometimes in meaning. For example (أشكال) and (إشكال), the first word means figures and second means problem.

8. The individual letter (و) which means "and" in English is often misused. It is a separate word and should have white space before and after it, but most of the time Arabic writers do not use the white space. This is a particularly common instance of the connected word problem.

9. The Arabic language contains a number of similar letters like Alef (ا) and the number 1 (1), and also the full stop (.) and the Arabic number zero [28].

10. Similarity of the characters: Recognizing differences between some characters is quite complicated. For examples, characters such as "ت" and "ن", "خ" and "غ", "ف" and "ق" are sometimes hard to differentiate.

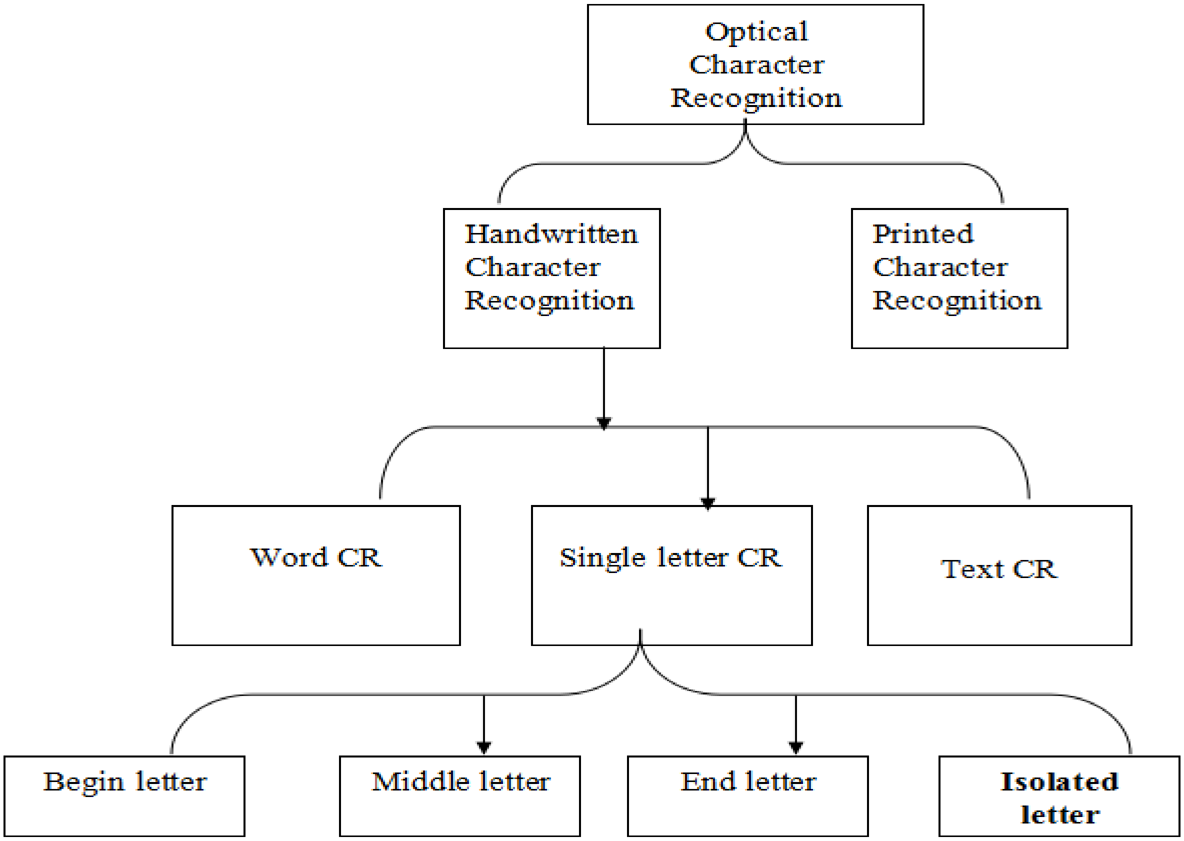

Optical Character Recognition
Handwritten Character Recognition
Printed Character Recognition
Word CR
Single letter CR
Text CR
Begin letter
Middle letter
End letter
Isolated letter

Figure 2. 5: The different families of Arabic character recognition.

2.4.6 System Structure of Arabic Handwritten Recognition

The sections below will explain the Arabic handwritten character recognition stages, which goes through several basic processes. These processes are used to transform the character into primary forms, preprocessing, feature extractions and to be recognized.

2.4.6.1 Image Acquisition

In Image acquisition, the primary purpose of the initial treatment is to put the character image into an approved and recognizable form. The recognition system acquires a scanned image as an input image. The image should have a specific format such as JPEG, BMT, etc. This image is acquired through a scanner, digital camera or any other suitable digital input device.

2.4.6.2 Preprocessing Techniques

The main aim of preprocessing is to enhance the inputted signal and to represent it in a way which can be measured consistently for robust recognition. A collection of different preprocessing techniques will be presented in this chapter section 2.5.

2.4.6.3 Features extraction

Before the recognition stage, some set of features of the image will be extracted. These features are a reduced representation of the contents of the image which focuses on preserving the characteristics that are most relevant to the task of recognition while eliminating those characteristics that are irrelevant to, or may confuse, the discrimination power of the recognition stage. The features extracted as a vector values are then passed to recognizer [29]. Some of features extraction techniques will be presented in this chapter section 2.6.

2.4.6.4 Character Recognition

The features vector obtained at the feature extraction stage are assigned a class label and recognized using a different method. In this stage the data set is divided into training set and testing set for each character and group. The classifiers used in this dissertation, are neural networks with back propagation to determine the similarity between characters. Adaptive neural fuzzy system (ANFIS) will also be used for distinguishing between the groups and characters classes. These classifiers will be discussed in this chapter section 2.7.

2.4.6.5 Post-Processing

Post-processing is the last activity to occur in a series of OCR processing stages. The goal of post-processing is to detect and correct misclassification in the OCR output.

The post-processor uses output of the classifier to decide on the recommended action. However, sometimes-different actions can have different costs. These costs can be taken into account when designing an optimal classification system. The post-processor can also exploit the context or combine results of several classifiers [30].

2.5 Preprocessing

The application process of preprocessing may vary in detail for different recognition systems. In this section, the techniques of the preprocessing process applied in the HACR system of this research are described.

Prepossessing is a basic task that precedes the tasks of image representation and recognition. Its importance is derived from the fact that the discrimination power is directly proportional to the digital image quality. With the highest image quality, then less confusion can be made about robust classification. Some of the common operations performed as preprocessing are as follows:

2.5.1 Morphological Image Processing

Is important technique, it is use for extracting image structure that are useful in the representation and description of region shape.

The basic morphological algorithms are: boundary extraction, region filling, and extraction of connected components, thinning and thickening.

The most common morphological operations used are:

A. Dilation, where the value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood.

B. Erosion, where the value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood.

C. Opening, this smooth contours and eliminates small islands and sharp peaks.

D. Closing, which smooth contours, fuses narrow breaks and eliminates small holes.

In the OCR field, the previous operations have been dealt extensively for handwritten or machine printed documents.

2.5.2 Thresholding

The task of thresholding technique is to extract the foreground (ink) from the background. Following scanning, character images are initially stored in a grey-level format. This means that the intensity of each pixel in the image may vary between a value of 0 and 255, where the value zero indicates a white pixel.

2.5.3 Noise Removal

Noise in the data typically originates from the limiting accuracy of the digitizing process, erratic hand motion and the inaccuracies of pen-down indication. A number of techniques are used for noise reduction. These include:

a. Smoothing: A common technique averages each point with its neighbors. Another well-known approach is to approximate the underlying ink trace to some standard curves.

b. Filtering: Eliminates duplicate data points and reduces the total number of points. The form of filtering depends on the recognition method. One filtering technique forces a minimum (or fixed) distance between consecutive points. When the writing is fast, however, the distance between successive points may far exceed the minimum. In this case interpolation can help obtain equally spaced points (called resampling). Another filtering technique produces more points in regions of great curvature.

c. Wild point correction: Replaces or eliminates occasional spurious points, typically caused by hardware problems.

d. De-hooking: Eliminates hooks that occur at the beginning and end of strokes, caused due to inaccuracies in pen down and pen up positions.

2.5.4 Normalization

Normalization refers to such operations as: the estimation and correction of a character's slant, the increase in the size of the character to a uniform size and may also reduce the character to the skeleton so that the line width is uniform to one unit wide. A number of algorithms are used for normalization such as the following:

a. De-skewing: Corrects slant. This can be applied at the letter or word level.

b. Baseline drift correction: Orients the word relative to a baseline.

c. Size normalization: Adjusts the letter size to a standard size.

d. Stroke length normalization: Forces the number of points in a stroke to a specified number for easy alignment [31].

2.5.6 Cropping the Image

Cropping refers to the removal of the outer parts of an image to improve structure. For example, omitting the unimportant areas from a photographic of the image or eliminate all the unwanted parts that are called noise. This process makes the feature more accurate and accordant.

Cropping technique stage: first the image is scanned row by row. The last set of rows from the top and the last set of rows from the bottom with all pixels having a value of "255" are obtained. In this process cropped the top and bottom part of the image. In the second step crop the left and right side of the image. After that the image is scanned column by column, the last column from the left with all pixels having a value of “255” is obtained. The same is done to the right side of the image.

2.6 Features Extraction [FE]

In this section described overview of FE process, history of FE, features types and PCA technique.

There are two main types of FE approaches; statistical and structural approaches of pattern recognition systems. Each type employs different techniques of features within the recognition and classification tasks which constitute a pattern recognition system. Statistical pattern recognition draws from established concepts in statistical decision theory to distinguish among data from different classes based upon quantitative features of the data [32]. Features approaches as following:

2.6.1 Structural features

Structural features can be obtained of Characters with high tolerance to distortions and style values. The characteristics represent on the characters in many forms. Structural features are based on the topological and geometrical properties of the character, such as aspect ratio, end points, loops, dots, branch points, junctions, strokes and their directions, inflection between two points, horizontal curves at bottom or top, etc. [32].

2.6.2 Statistical Features

Statistical pattern recognition draws from constructed concepts in statistical decision theory to distinguish among data from different classes based upon quantitative features of the data. There are wide types of statistical techniques that can be used within the description task for feature extraction, ranging from simple descriptive statistics to complex transformations [32]. The following shows the major statistical features used for character representation:

1) **CCOB Features:** Vertical and horizontal crossings are found by counting the number of white-Black -white transfers when scanning the image's pixels on a vertical line and a horizontal line, respectively [33]. Outliers and blank ink histogram [34].

Cross feature Count the number of transitions from background to foreground pixels along vertical and horizontal lines through the character image. **Cross feature** Count the number of transitions from background to foreground pixels along vertical and horizontal lines through the character image as shown in Figure 2.6. **Outliers** (Right, Left, Top and Down) - calculate the distances of the first image pixel detected from the upper and lower boundaries of the image along the vertical lines and from the left and right boundaries along the horizontal lines. **Black ink histogram** is calculate the black ink histogram features as shown in figure 2.7.

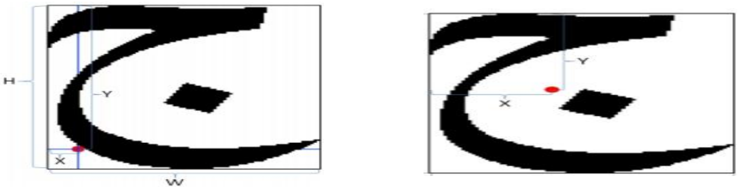


Figure 2. 6: Vertical and Horizontal character, crosshair and center of mass.

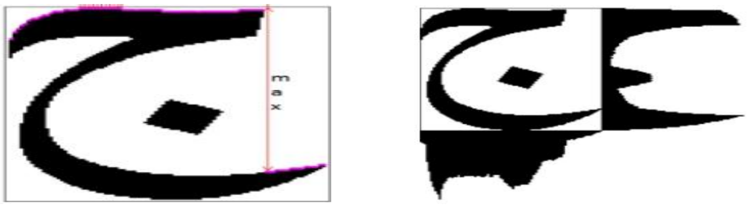


Figure 2. 7: Right, Left, Top and Down character, Outliers and Blank ink.

2) **Principal Component Analysis (PCA):** The principal component analysis is one of the oldest and most famous of the multivariate analysis techniques. Put the basis first by Pearson (1901), and developed by Hoteling (1933). However, it has not been used widely until the advent of electronic computer, but now use in the statistical software package [35].

The main idea of using PCA for character recognition is to express the large one dimension vector of pixels produced from two dimension character image into the compact principal components of the feature space [36]. PCA requires a mathematical method to turn correlated variables into disjointed variables [37].

The main objective of PCA is to reduce the large dimensionality of the data space to the smaller basic dimensionality of feature space (independent variables) [37].

2.7 Classifications Methods

This section provides background on artificial Neural Networks (ANN), fuzzy logic and neuro-fuzzy (NF) systems. Neuro-fuzzy systems are a combination of neural network and fuzzy logic. The integration of fuzzy logic and neural networks produces different structures. The NF system used will be explained in detail last in this chapter.

2.7.1 Artificial Neural Networks (ANNs)

In this subsection presented, introduction to artificial neural networks, ANNs architectures, transfer function, learning methods and back-propagation.

2.7.1.1 Introduction

An artificial neural network is a correlated gathering of nodes, which are an electronic modules based on the neural structure in the brain. This brain modeling used as a guide to develop machine solutions. Artificial Neural Network (ANN), are computers whose architecture is modeled after the brain. ANN consists of large number of simple processing units which are connected together in a complex communication network. It is a device with many inputs and one output [38]. Some of them defined a neural-network structure “as a collection of parallel processors connected together in the form of a directed graph, organized such that the network structure lends itself to the problem being considered” [39].

A typical multilayered feedforward neural network [40] is shown in Figure 2.8. It consists of massively interconnected simple processing elements (“neurons” or “nodes”) arranged in the form of several layers, and each layer is connected with the other by links of weights which specifies the power of the link between the layers; these weights are the internal parameters of the network. The input neurons are connected to the output neurons through layers of hidden nodes. Each neuron receives signals in the form of inputs from other neurons. Figure 2.9 shows simple model explains the behavior of a neuron.

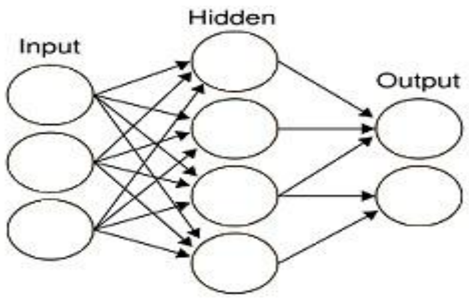


Figure 2. 8: Multilayered feedforward neural network.

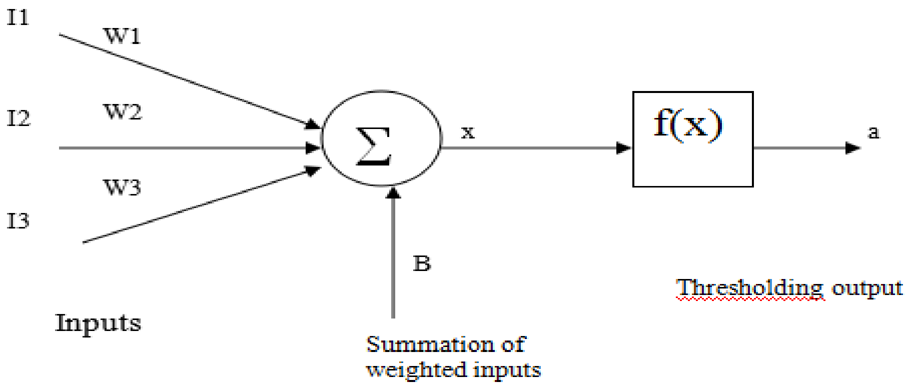


Figure 2. 9: Simple model of an artificial neuron.

Where ***I*₁**, ***I*₂**, and ***I*₃** are the inputs, ***W*₁**, ***W*₂**, and ***W*₃** are the weights, ***B*** is the bias, **x** is the intermediate output, and ***a*** is the final output. The equation for **a** is given by

$$\boldsymbol{a = f\ (W_1I_1+W_2I_2+W_3I_3+B)} \qquad (2.1)$$

Where *f* could be any function. Most often, *f* is the sign of the argument (i.e. 1 if the argument is positive and -1 if the argument is negative), linear (i.e. the output is simply the input times some constant factor), or some complex curve used in function matching (not needed here).

2.7.1.2 Main Architectures of Artificial Neural Networks

In general, an artificial neural network can be divided into three parts, named layers, which are as follows:

(a) Input layer

In this layer the input values for signals or features are received from the external fields. The activation functions normalize the input values to a specific value. The new value will pass to the next layer.

(b) Hidden Layers

These layers located between the input and output layers are considered the main processor for neural network processes. They are composed of neurons which are responsible for extracting patterns associated with the process or system being analyzed.

(c) Output layer

This layer is also composed of neurons, and thus is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers [41].

2.7.1.3 Transfer Function

The way an ANN works depends on the weights and the input-output function (transfer function) that is specified for the units [42]. This function can be categories into three types as shown in figure 2.10.

1- Linear function

The linear activation function multiplies the net input with a constant value k to produce the neuron output. Also note that the range of the linear activation function is (-1; 1).

2- Threshold

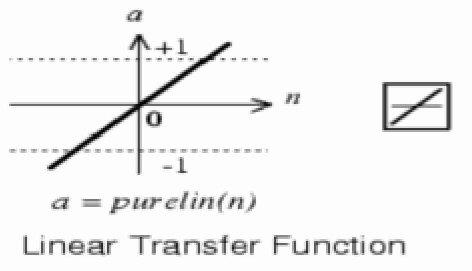
The threshold function is a binary valued function with the range [0, 1]. The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value. There also exists threshold functions with other ranges (e.g. [-1; 1]).

$$\varphi(u) = \begin{cases} 1 & : u \geq 0 \\ 0 & : u < 0 \end{cases} \qquad (2.2)$$

3. Sigmoid function

A sigmoid function with the range (0, 1). The parameter a affects the slope of the function and when a goes towards infinity, the sigmoid function becomes a threshold function.

$$\varphi(u) = \frac{1}{1 + e^{-au}} \qquad (2.3)$$



Linear Transfer Function

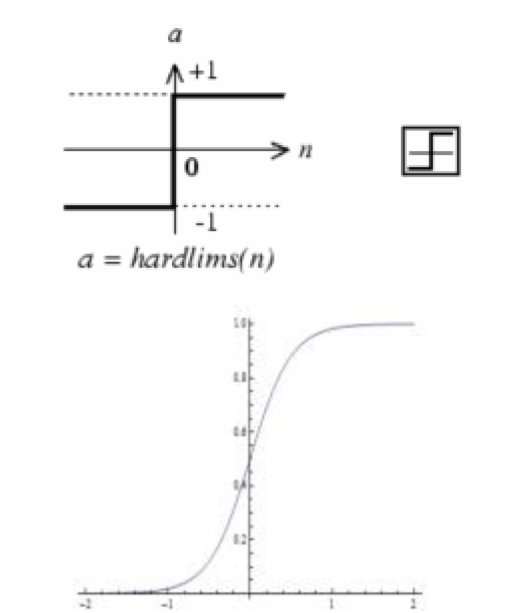


Figure 2. 10: Illustrates transfer function types. (Linear, threshold and sigmoid)

2.7.1.3 Neural Network Types

There are different types of artificial neural networks, and they come with different number of layers and the way the layers are interconnected. The types can be divided as follows: (i) single-layer feedforward network, (ii) multilayer feedforward networks, (iii) recurrent networks and (iv) mesh.

2.7.1.4 Types of Neural Networks According to Nutrition

There are two types of neural networks based on the method of nutrition which are feedforward and backforward. In the first type of network the data move in one direction only from the input layer to the hidden nodes [42]. The output from one layer of neurons feeds forward into the next layer of neurons. There are never any backward connections. In the second, a feedback network allows output to be inputted to preceding layers. It is a process whereby the modifications to the weights are controlled by the activity of some of the products of the outputs. This type of network is also known as the recurrent network [42].

2.7.1.5 Learning Methods

There are many learning methods for neural network. These methods will be used for regulation of the network weight so that the learning method will cause the output to gradually converge to a value such that the input is the desired output. Learning methods can be dividing into three basic types; supervised, unsupervised, and reinforced [56].

The unsupervised techniques are learning method, which means no "teacher" during the training process. The system in unsupervised learning has no knowledge about the correct outputs of the corresponding input learning. The supervised learning is process under the supervision of "teacher" who monitors the learning outputs. The supervised learning (SL) is method to minimization the error between the desired outputs and actual outputs. The aim is to determine a set of weights which minimizes the error. SL is common to many learning methods in which the best known is the least mean square (LMS) [42]. An example of supervised learning is the back-propagation algorithm which is one of the most popular and prevalent training methods used in neural networks.

2.7.1.6 The Back-Propagation Neural Network (BPNN)

The back-propagation algorithm used with neural networks is based on the gradient descent for learning parameters of these surfaces [43].

Back-propagation algorithm consist of three layers of nodes; the first layer named the input layer, the second layer named the hidden layer which consist of at least one layer and the third is named the output layer. The output from the input layer is connected as an input into the hidden layer and the output value from the hidden layer is transmit into the output layer as an input. The output layer gives the final output of the ANN. In back-propagation neural network, the network produces the input pattern from layer to layer until the output pattern is generated by the output layer. If there is a difference between actual and desired output patterns, repeat the process of learning until the weights are adjusted to minimize this error. At the end of any learning process propagated backwards through the network from the output layer to the input layer and so on [44].

2.8 Fuzzy Inference System

This section presents information about the theory of fuzzy logic, presented definition of a fuzzy set, explained fuzzy and member functions.

2.8.1 Introduction

Fuzzy Logic (FL) was first introduced by Lotfi Zadeh [45]. It was developed for solving decision making problems using "IF-THEN‟ rules. Later, it was used to deal with uncertain and imprecise data management. It has been applied in several areas such as automated control, consumer products, industrial systems, automotive, decision analysis, medicine, geology, controlling aircraft flight, chemical reactor and processes, and applications of artificial intelligence such as expert systems, pattern recognition and robotics [6]. FL produces new models for dealing with natural languages and knowledge representation such as precipitated natural language, theory of hierarchical definability unified theory of uncertainty [46]. The real values of the assumptions indicate the degree of emphasis on the premise that is correct. Fuzzy logic is an important recognition method and has been used as a tool of interpreting vague, incomplete, ambiguous, imprecise and noise information [47]. There are many motivations that prompted developers to present fuzzy logic with the development of computer software. FL can create a desire to design a computer programs that can mimic the human mind. There are many concepts that are part of fuzzy logic such as fuzzy set theory, fuzzy logic operations (complement operation, union operations and the intersection operation) these operations can be also applied on binary sets [48].

2.8.2 Fuzzy Set and Membership Function

The concept of fuzzy set allows partial membership of the element of a set. Membership of an element of fuzzy set is expressed by the degree of compatibility of the element with the definition of that set.

Membership may be expressed as a value between number 0 (complete non-membership) and 1 (complete membership). The membership value between 0 and 1 represents the partial membership concept [49].

A fuzzy set A in the universe of discourse X (into the interval [0, 1]) can be represent.as follows:

$$A = \{(x, \mu_A (x)) \mid x \in X \} \qquad (2.4)$$

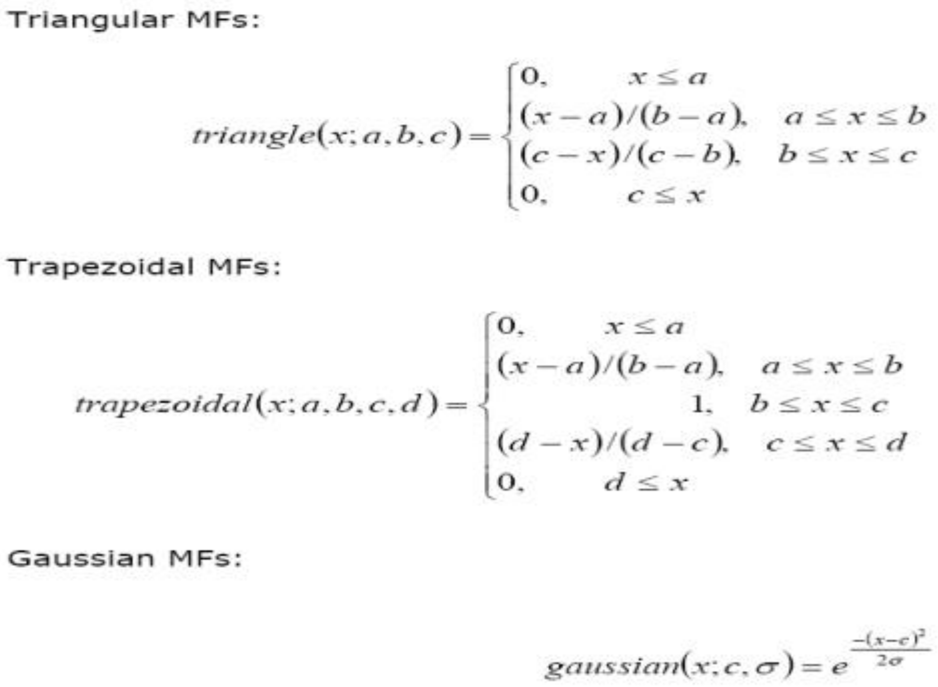
Where $\mu_A(x)$ is called membership function for the fuzzy set *A*. It maps each x to a membership grade between 0 and 1.

Generally, there are many common shapes of membership function Examples of membership functions Triangular, Trapezoidal and Gaussian can be seen in Figure 2.11 and described with the formulas for any function.

Figure 2.12 shows some of general membership function curve, Triangular, Trapezoidal and Gaussian member functions, respectively.

2.9 Neuro-Fuzzy

The previous section briefly presented the concept of the two main components, ANN and FL building up a Neuro-Fuzzy system individually. This section gives a description on Neuro-Fuzzy system and the different architectures can be discussed to show how different approaches managed to combine ANNs with Fuzzy Systems. ANFIS architecture and Takaji-Sugeno are presented.



Triangular MFs:

$$triangle(x;a,b,c)=\begin{cases}0, & x\le a\\(x-a)/(b-a), & a\le x\le b\\(c-x)/(c-b), & b\le x\le c\\0, & c\le x\end{cases}$$

Trapezoidal MFs:

$$trapezoidal(x;a,b,c,d)=\begin{cases}0, & x\le a\\(x-a)/(b-a), & a\le x\le b\\1, & b\le x\le c\\(d-x)/(d-c), & c\le x\le d\\0, & d\le x\end{cases}$$

Gaussian MFs:

$$gaussian(x;c,\sigma)=e^{\frac{-(x-c)^2}{2\sigma}}$$

Figure 2. 11: Illustrates transfer function. (Triangular, Trapezoidal and Gaussian)

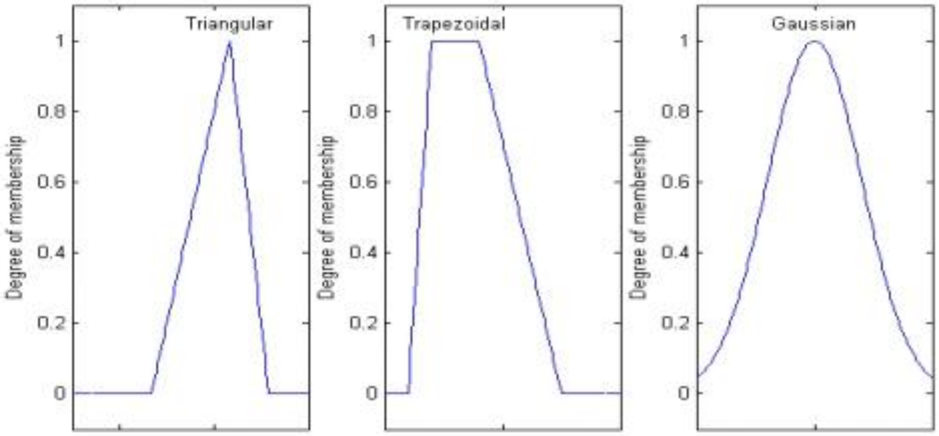


Figure 2. 12: Examples of membership functions.

2.9.1 Introduction

A neuro-fuzzy technique called Adaptive Network fuzzy Inference system (ANFIS) is a combined system that integrates the learning abilities of ANN and excellent knowledge representation and inference capabilities of FL [50]. ANFIS has been big power to be an effective tool for adjusting the membership functions of fuzzy inference system. The objective of an ANFIS [50] is to integrate the best features of Fuzzy Systems and Neural Networks.

Generally, ANFIS is a feed forward network in which each node performs a particular function on signals comes from previous.

2.9.2 Types of Neuro-Fuzzy Systems

In general, all the consolidation of techniques based on neural networks and fuzzy logic can be called neuro-fuzzy systems [51]. The different combinations of these techniques can be divided into three classes; Cooperative Neuro-Fuzzy System, Concurrent Neuro-Fuzzy System and Hybrid Neuro-Fuzzy System

2.9.3 Hybrid Neuro-Fuzzy System

In this category, a neural network is used to learn some parameters of the fuzzy system (parameters of the fuzzy sets, fuzzy rules and weights of the rules) of a fuzzy system in the implementation processes. There are several different models to develop hybrid neuro-fuzzy systems. These models are similar in its general conceptual, but they present basic differences. The popular model used is Adaptive Neural Fuzzy Inference System, which produces in next subsection.

2.9.4 Adaptive Neuro Fuzzy Inference System (ANFIS)

ANFIS [52] implements an FIS model and has a five layered architecture as shown in Figure 2.13. The first layer received inputs from external environment and the input variables are fuzzified. The second layer computes the rule antecedent part. The third hidden layer normalizes the rule strengths followed by the fourth hidden layer where the consequent parameters of the rules are determined. The last layer is the output layer which computes the summation of all input signals. ANFIS uses back-propagation learning to adjust the premise parameters and Least Mean Square estimation to control the consequent parameters. A step in the learning procedure has got two parts: In the first part, the input patterns are propagated, and the optimal consequent parameters are estimated by an iterative least mean square procedure, while the premise parameters are assumed to be fixed for the current cycle through the training set. In the second part the patterns are propagated again, and in this epoch, back-propagation is used to modify the premise parameters, while the consequent parameters remain fixed. This procedure is then iterated.

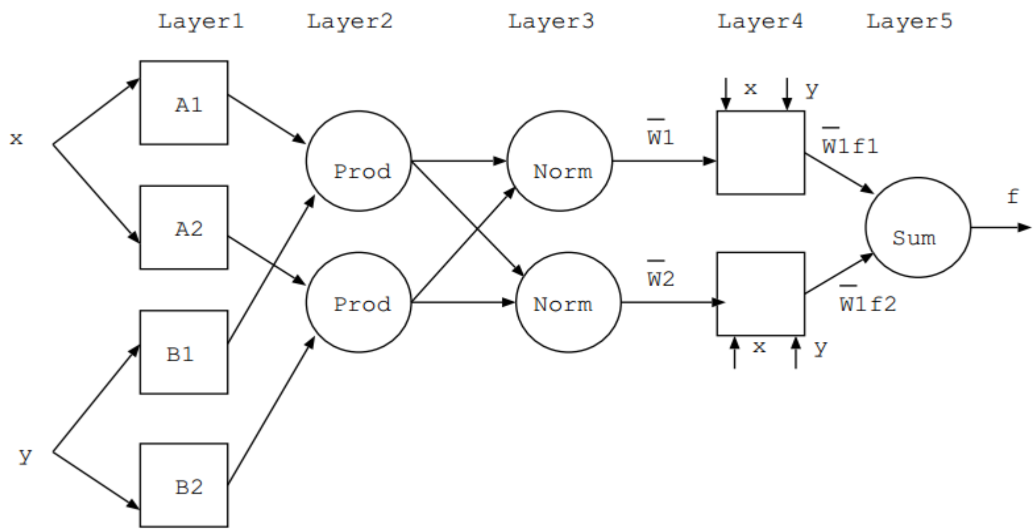


Figure 2. 13: ANFIS architecture for a 2-inputs Sugeno fuzzy model.

2.9.4.1 ANFIS Architecture

ANFIS is a structural of network representation of Sugeno-type fuzzy systems merged with the neural learning capabilities. The network is consistent of nodes with specific functions collected in layers. ANFIS is able to construct a network realization of IF / THEN rules [50]. Figure 2.13 shows an example for an ANFIS structure.

Example: A two inputs (*x* and *y*) and one output (*f*) ANFIS

Rule 1 : IF *x* is $\boldsymbol{A_1}$ and *y* is $\boldsymbol{B_1}$, then

$$f_1 = p_1x+q_1y+r_1 \tag{2.5}$$

Rule 2 : IF x_1 is $\boldsymbol{A_2}$ and x_2 is $\boldsymbol{B_2}$, then

$$f_2 = p_2x+q_2y+r_2 \tag{2.6}$$

Layer 1: adaptive nodes, is the input layer. Each neuron in this layer transmits external crisp signals directly to the next layer. That is,

$$\boldsymbol{O_{1,i} = \mu_{Ai}(x) \quad i=1,} \tag{2.7}$$

$$\boldsymbol{O_{1,i} = \mu_{Bi-2}(y) \quad i=3,\ 4} \tag{2.8}$$

Where $\boldsymbol{\mu_{Ai}(x)}$ and $\boldsymbol{\mu_{Bi-2}(y)}$ are the degree of membership functions for the fuzzy sets *Ai* and *Bi*, respectively,

$$\mu_{A_i}(x) = \frac{1}{1+[\frac{(x-c_i)^2}{a_i^{\,2}}]^{b_i}} \tag{2.9}$$

Where {*ai*, *bi*, *ci*} are the parameters of a membership function that can change the shape of the membership function. The parameters in this layer are typically referred to as the premise parameters.

Layer 2: fixed nodes with function of multiplication, is the fuzzification layer. Neurons in this layer represent fuzzy sets used in the antecedents of fuzzy rules. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron’s fuzzy set.

$$\boldsymbol{O_{2,i} = w_i = \mu_{Ai}(x)\times\mu_{Bi}(y) \quad i=1,\ 2} \tag{2.10}$$

(the firing strength of a rule)

Where ***wi*** is the output that represents the firing strength of each rule.

Layer 3: fixed nodes with function of normalization, is the fuzzy rule layer, which are used to calculate the normalized firing strength. The output of this layer can be represented as:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2} \qquad i = 1, 2 \tag{2.11}$$

(Normalized firing strength)

Layer 4: In layer 4, the nodes are adaptive. The output of each node in this layer is simply the product of the normalized firing strength and for a first order Takagi-Sugeno model. The output of this layer can be described as:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \tag{2.12}$$

where {p, q, r} is a set of consequent parameters which can be identified using the Least Square Estimation (LSE).

Layer 5: a fixed node with function of summation is the defuzzification layer. Each neuron in this layer represents a single output of the neuro-fuzzy system. It takes the output fuzzy sets fuzzy system. It takes the output fuzzy sets clipped by the respective integrated firing strengths and combines them into a single fuzzy set.

$$O_{5,1} = overall\ output = \sum_i \overline{w}_i f_i \tag{2.13}$$

2.9.4.2 Hybrid Learning Algorithm (HLA)

This algorithm is an integration of two functions, the gradient descent and the least squares methods. The objective of this combination is to minimize the error in the learning stage. The HLA consist of two passes known as forward and backward passes as shown in table 2.2. In the forward pass, the info flows forward until o4 j and the consequent parameters are determined by the least square approach. In the backward pass the error signals is propagate and gradient descendent approach used to update the premise parameters. This process is shown in Figure 2.14 Hence; HL approach is much faster by reducing the search space dimensions of the back-propagation method [50].

When the premise parameters are fixed:

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \\ &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 \\ &\quad + (\overline{w}_2 x) q_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2 \end{aligned} \tag{2.14}$$

where p1, q1, r1, p2, q2, and r2 are the linear consequent parameters.

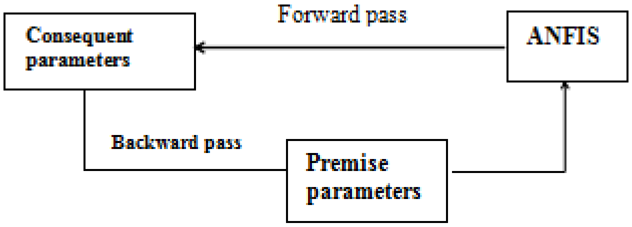


Figure 2. 14: Forward and backward pass in the hybrid algorithm.

Table 2. 2: Two passes in the hybrid learning procedure for ANFIS.

Parameters	**Forward pass**	**Backward pass**
Premise	Fixed	Gradient descent
Consequents	Least- squares estimator	Fixed

2.9.4.3 Takagi Sugeno Fuzzy Inference System (TS-FIS)

TS models are a powerful practical engineering tool for modeling and control of complex systems. The output of a TS model can be linear or a constant. As ANFIS is based on Takagi–Sugeno fuzzy inference system [50]. A typical rule in a Sugeno fuzzy model has the form following:

IF x is A AND y is B THEN z is f (x, y)

Where x, y and z are linguistic variables, A and B are fuzzy sets on universe of discourses X and Y and f (x, y) is a mathematical function. In ANFIS, the output linear membership functions of the consequent part of TS-FIS are automatically adjusted.

2.10 Summary

The topics discussed in this chapter are as follows:

1. This chapter introduced a review study of pattern recognition with definitions, applications, pattern recognition methods.

2. An introduction to OCR, types of OCR and families of Character Recognition.

3. An overview of the written Arabic characters, AHCR characteristics and AHCR applications. The problems are occurring when recognizing Arabic handwritten characters.

4. The pre-processing stage with its inclusive steps which include Document analysis, Binarization, cropping, normalization and morphological.

5. The feature extraction phase with samples of common algorithms. The algorithms presented are **S**tatistical Features and structural features. In addition, we focused on CCOB, PCA and features.

6. Classification phase with samples of some classification methods. This chapter produced neural network method, activation function, learning algorithm and the Back-Propagation algorithm were provided. Therefore, presents the basics of fuzzy set theory and explained the theory behind Neuro-Fuzzy systems, the types of neuro-fuzzy, present's architecture of Adaptive Neuro Fuzzy Inference System (ANFIS) and Hybrid Learning Algorithm.

CHAPTER III

LITERATURE REVIEW

The chapter presents the review of the research that has been done in the field of handwritten Arabic characters recognition that is related to this dissertation. In the first section, studies on preprocessing techniques used for Arabic character recognition, in the second section the applied features extractions are presented. In the third section, related works on recognition and classification methods development and implementation are given. At the end of each section, a summary is presented in a table format. In the fourth section, work related to Neuro-Fuzzy Algorithms and Handwriting is presented. In the last section, studies on multi-stage classification are presented.

3.1 Overview

Character recognition goes through several stages as shown in figure 3.1. The first stage deals with scanning the text page as an input. There are some preprocessing steps followed after the scanned page to improve the performance of letter recognition. The following are some of the preprocessing techniques: resized, noise removal, median filter and grayscale, skew detection and correction, etc. After preprocessing, the text image will be segmented into many shapes, lines, words, sub-words, characters or sub characters. The next stage is to extract features. Features are usually used as inputs to train the classifier to build the models (training and testing) through the classification stage. The last stage in the recognition system is the post-processing stage which is used to improve recognition.

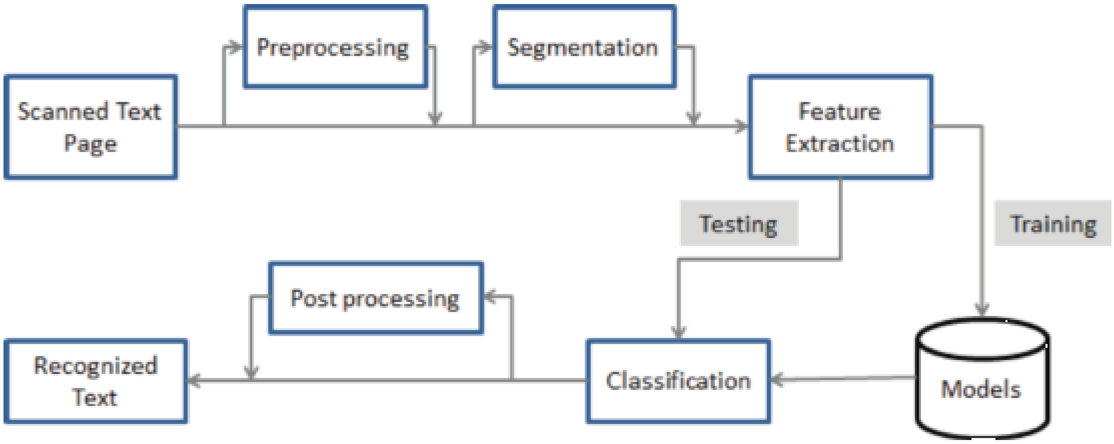


Figure 3. 1: Character recognition stages.

3.2 Preprocessing

Preprocessing stage involves series of techniques performed to enhance the image to make it suitable for segmentation. The preprocessing techniques that will be used are the following: noise removal, normalization, cropping, morphology, skew correction, thinning and slant. The aim of preprocessing stage is the removal of all elements in the word image that are not useful in the recognition process.

The research that is related to the preprocessing stage for Arabic handwritten text images can be classified as follows:

1. Thresholding

This technique involves transforming an input image from a gray image into a binary image and was cited by [55, 56, 65, 67, 74 and 81] to get a good features from the image.

2. Thinning

The thinning technique will transform the binary image of the character to a one pixel thick image. In [53] and [70] they have used this technique in their researches.

3. Normalization

Normalization is the process to converting the random size for images into standard size. The research that was done in [56] used a common defined size of 30x30 pixels matrix.

4. Noise Removal

The major objective of noise removal is to remove any unwanted bit-patterns which do not have any significance in the output. The Median filtering technique was use by [54, 56, 59, 63 and 75] to remove noise.

5. Skew detection and skew correction: This technique used by [54 and 59].

6. Skeletonization: Used by [54 and 70].

7. Smoothing: Used by [57, 67, 72 and 74].

8. Filterization, resize and cropping

Filterization and resize used by [56 and 65] and cropping used by [61].

9. Horizontal Projection Profile: Used by [53 and 60].

3.3 Feature Extraction

Feature extraction is the heart of any pattern recognition application. The traditional goal of the feature extractor stage is to characterize an object by making numerical measurements. Good features are those whose values are similar for objects belonging to the same category and distinct for objects in different categories. Feature extraction is problem dependent [30]. Features are extracted from the image of a word/character which is expected to represent the shape of the images. There are a number of famous features that are related to the character recognition problem. There are numerous types of features proposed by researchers. There are two types of features; Structural features such as loops, branch points, endpoints, dots, etc. and Statistical features such as pixel densities, histograms of chain code directions, center of mass black ink histogram.

3.3.1 Structural features

These types of features were used by [53, 62, 68, 71 and 78]. In [53] structural features that were used are the following: the length of the character, the width of it, if it has a loop or not, if there is a right character to connect to it, if there is a left character to connect to and if there is a complement of the character like the zigzag shape (Hamza), one dot, two dots or three dots. In [62] they used features such as loops, holes, strokes, vertical lines and cusps.

3.3.2 Statistical features

Many statistical features were used for pattern recognition. Pixel densities features were used in [4, 58 and 61]. Sum of black and white pixels were used in [58] and in [61] detecting black and white points were used as statistical features. Moment invariants were used by [54 and 73]. In [54] they proposed seven invariant moments such as invariant under reflection, rotation and scaling. In [73] they used scale, orientation and center of gravity. Vertical and horizontal projections were used in [53 and 60]. In [53] the longest spike which represented the baseline was used. In [70] branch, start-point and end-point of a character features were used. Wavelet transforms features were used by [75 and 76]. In [55 and 64] they used the following features: determines the body and secondary part, position of the part above or below, loop and Radon transform of the characters were used. Code of chain feature used in [69] and Pseudo-Zernike Moments, size, rotation, translation invariant features were also used in [81] and center of mass, crosshair, outliers and black ink histogram features are used in [95,96]. Table 3.1 summarizes some of the preprocessing techniques and features extraction methods used in offline Arabic text recognition.

3.4 Classification Approaches

Classification is the main stage of AHCR system. It uses the features in the feature extraction stage as inputs to the model to identify the text segment according to stated rules. These include Hidden Markov Model (HMM), Support Vector Machines (SVM), Artificial Neural Networks (ANN), k-Nearest Neighbors (k-NN), fuzzy logic, genetic algorithm, neuro-fuzzy and others. The different approaches for classify character recognition will presented in this section.

Table 3. 1: Summary of some of the preprocessing techniques and features extraction methods used in the offline Arabic text recognition.

Preprocessing	Features	Reference, date
	Sum of black and white pixels.	[4], 2010
Thinning technique for Arabic words and finds the vertical and horizontal projection profile.	Length of the character, the width of it, has a loop or not, a right character to connect to it, if there is a left character to connect to and if there is a complement of the character like the zigzag shape (Hamza), one dot, two dots or three dots.	[53], 2010
Noise analysis and removal, Skew detection and correction and Evaluating the gap between the words and characters.	The features in each window are extracted through seven invariant moments example of these moments are invariant under reflection, rotation and scaling.	[54], 2011
Binary, filtering, and skeleton.	Divide the letters into regions for determine the body and secondary part, position of the part above or below, loop and Radon transform.	[55], 2011
Filter, binary image and resized.	Width of the dynamic-sized window and the height.	[56], 2012
Smoothing and de-	Different the values, similar value	[67], 2009

noising.	and Independence features.	
	Width, length, number of pixels and height-to-width ratio.	[58], 2005
Removal of isolated pixels, skew detection and correction.	Invariant under image translation, scaling and rotation.	[59], 2003
Vertical and horizontal projection.	Horizontal and vertical projection profile and Laplacian filter.	[60], 2012
Convert RGB image and cropping the white area around the word.	Detecting black and white points.	[61], 2012
Black and white format and noise.	Loops, holes, strokes, vertical lines, and cusps.	[62], 2008
	Genetic algorithms representing a subset of evolutionary computation techniques.	[63], 2008
	Secondary components, Main body, center of mass and crossings feature.	[64], 2008
Resizing, filtration, and converting to binary.	Ratio of white pixels to black pixels, ratio between the two farthest pixels, average of the spatial segment, variance of the farthest two vertical pixels, average value of the line and total variance of the segment pixels.	[65], 2012
	Aspect and stroke ratio.	[66], 2009
Binarization and smoothing.	Discrimination, reliability and independence	[67], 2001
Binary images.	The contours (dots and holes).	[68], 2006
	Morphological primitives and the	[69], 2006

	statistic primitives.	
Skeletonization and thinning	Branch and start-point and end-point of a character or an end-point and start-point of a straight piece in a character.	[70], 2009
	Number of dots in the character and where the dot is, if it's up, down or in the middle of the character.	[71], 2010
Smoothing operation and cursive characters.	The velocity profile of each stroke.	[72], 1997
	Moment invariants, scale, orientation and center of gravity	[73], 2007
Binary and smoothing.	Uniform and no-uniform.	[74],2006
Noise reduction and edges detection.	Hough transform and wavelets transform.	[75], 2006
	Statistical features from the standard deviation of the wavelet coefficients besides the density of black pixels.	[76], 2006
	PCA for feature extraction, calculate the Eigenvectors and Eigen values.	[77], 2007
	Structural features and Direction feature.	[78], 2013
	- Dimensions of the including rectangle (high and width). - Relative position of the including rectangle - Positions of the references points. - Direction of the trajectory on the level of the point's references.	[79], 2009

	Velocity profile of each stroke.	[80], 1997
Binarization and thinning	Pseudo-Zernike Moments, size, rotation and translation invariant.	[81],1996
	Density, Aspect Ratio and Character Alignment Ratio Clustering	[30]
	dots, position, and where it is a dot or zigzag	[82]
	bitmaps technique and modified chain-code direction frequencies	[83]

3.4.1 Hidden Markov Models (HMM)

Many researchers have used (HMM) for Arabic text recognition [74, 75 and 79]. The general trend of using HMM is to use a sliding window of the text line image to convert a 2-dimensional image to a 1-dimensional feature vector [82]. In [74] they used a HMMs algorithm with different explicit distribution for the state duration and after combining statistical and structural features the best recognition rate was 91.23%. In [75] they used wavelet transform features extraction and presented the edge detection of the character features with Hough Transform. In [79] they used genetic algorithm (GA) to optimize the sequences of handwritten strokes with HMM classifier.

3.4.2 Fuzzy logic (FL) approach

Fuzzy logic is a soft computing approach based on "degrees of truth" rather than the usual "true or false" logic. Many researchers have applied FL for recognition [4, 58, 69 and 73].

3.4.3 Neural network

The main driving force behind neural network research is the desire to create a machine that works similar to the manner in which our own brain works. Neural networks have been used in a variety of different areas to solve a wide range of problems. Unlike human brains that can identify and memorize the characters like letters or digits; computers treat them as binary graphics. Therefore, algorithms are necessary to identify and recognize each character [83].

Neural networks were used for Arabic text recognition [57, 63, 65, 70 and 78]. Back-propagation algorithm was used in [57] for recognition. In [63] they proposed two neural networks, a Multi-Layer Perceptron (MLP) and a Learning Vector Quantization (LVQ). In [65] they present cascaded networks to recognize the characters with 120 features for each character’s image. Back propagation Neural Network algorithm was also used in [70] with histogram features. In [78] authors have designed Radial-Basis neural network classifier, investigated and compared among results of four different artificial neural network models.

3.4.4 K-Nearest Neighbor

K-nearest neighbor algorithm (k-NN) [84 and 85] is a method for classifying objects based on closest training examples in the feature space. The K-NN algorithm is amongst the simplest of all machine learning algorithms; an object is classified by a majority vote of its neighbors with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small integer).

3.4.5 Neuro-Fuzzy

We found many researchers that have used Neuro-Fuzzy approach [62, 72, 76, 80 and 81]. In [55] they used Fuzzy logic with ART neural networks.

Table 3.2 shows a summary of some of the comparison of offline Arabic character recognition algorithms.

Table 3. 2: Illustrates summarizes some of the recognition algorithm with respect to the dataset and accuracy in offline Arabic text recognition.

Algorithm Type	Data set	Accuracy	Reference, date
Fuzzy logic	96 different shapes of characters.	88%	[4], 2010
Genetic algorithm	15000 handwritten words written by 100 writers.	87%	[53], 2010
Minimum distances and cosine (Ø)	Collected by forty subjects.	83.09%, 92.86 %.	[54], 2011
Multilevel classifier		{93, 84, 89 and 85%} Isolated, beginning, middle and end.	[55], 2011
Dynamic-size windowing	100 character shapes, from many articles in Alrai newspaper.	96%	[56], 2012
Back-propagation	100 letters	95%	[57], 2009
Fuzzy logic			[58], 2005
Radial-Basis Function RBF Network	149 characters	73%	[59], 2003
Production rule, NN and decision tree	504 characters for training, 336 for testing	97.6%, 97.5% and 98.8%	[60], 2012
Matched data with black and white pixels	IFN/ENIT	81%	[61], 2012

Fuzzy ART neural network	700 sample characters	95%	[62], 2008
Multi-Layer Perceptron (MLP), Learning Vector Quantization (LVQ) and K–Nearest Neighbor (KNN)	2000 instances	98%, 96.54 and 97.15	[63], 2008
Linear Discriminant Analysis (LDA)	104 characters	87%	[64], 2008
Cascaded neural networks	100 different separated characters	68.10%	[65], 2012
Semi-supervised	IFN/ENIT	67% and 77%	[66], 2009
based segmentation technique		90%	[67], 2001
Fourier Spectrum (MFS)	78 Arabic characters	95.6%	[68], 2006
Fuzzy Logic (FL) and Expert System (ES)	280 characters	98.97%	[69], 2006
Back propagation Neural network		98.7%	[70], 2009
Decision tree	336 samples	97.6%	[71], 2010
Fuzzy neural network	more than 3000 characters	89%	[72], 1997
Particle swarm optimization	448 samples	82%	[73], 2007
a HMMs algorithm	26459 Arabic words	91.23 %.	[74],2006

	written by 411 different writers		
Hidden Markov Models	170.000	96%	[75], 2006
Neuro-fuzzy	174394 Arabic characters in 9 different fonts.	95.64%.	[76], 2006
Nearest Neighbor	200 samples	90%.	[77], 2007
Radial-Basis neural	620 samples	95.32%	[78], 2013
Hidden Markov Model (HMM)		81% and 83%	[79], 2009
Fuzzy neural network	2000 characters written by the one writer	89%	[80], 1997
Neural networks and Fuzzy neural networks	3700 character samples.	99.85%.	[81],1996
multiple classifiers system		97%	[30]
LeNet NN			[82]
SVM classifier		96.68%	[83]
Neuro-fuzzy	Printed characters	97%	[87]
Fuzzy ART network	Handwritten characters	90.1	[88]

3.5 Neuro-Fuzzy Algorithms and Handwriting

The integration of neural networks and fuzzy logic inference systems could be formulated into three main categories; cooperative, concurrent and integrated neuro-fuzzy models. In this section we will present the neuro- fuzzy algorithms used for handwriting recognition systems.

In 2010, Majidaet al [4] proposed a system to recognize isolated Arabic characters using fuzzy logic approach. The stages of classification that they used were three. The first is sum of pixels stage in which they determine the sum of black and white pixels and saved these matrices in different files. The second stage is templates comparison in which matching between rows is performed by comparing every template against every other saved template. In the last stage they used the matrices as input to the fuzzy system [4].

In 2009, Albakoor et al [70] proposed two important concepts; segmentation on the basis of word histogram and baseline estimation. Preprocessing stage techniques that were used are Skeletonization and thinning. The segmentation technique that was used here depended on the histogram observations of lines and columns included in the histogram of the word. The words are segmented into their characters. Feature extracted with the following characteristic points: branch, start-point and end-point of a character or an end-point and start-point of a straight piece in a character. In classification they used Back propagation Neural Network algorithm, the recognition rate was 98.7% [70].

In 1997, Adel [72] described an on-line Arabic handwritten characters system. In this system, he used a fuzzy neural network for classifying characters. Preprocessing methods used were smoothing operation and cursive characters. The features extracted from each character are the neuro physiological parameters of the equation describing the curvilinear velocity of the script. The recognition stage is the Beta fuzzy neural network which has 100 neurons in the hidden layer and 55 neurons in output layer. The hierarchical neural network structure that he used contained six stages of three layers each. The system implemented in C/C++ language and achieved a recognition rate of 89%.

In 2006, [76] proposed an Arabic characters recognition system using wavelet transform to extract features. Authors have used a neuro-fuzzy approach for character recognition. They used Fuzzy logic as a tool for enhancing the ability to deal with the recognition problem. The purpose of fuzzy features was to map the extracted features to values from 0 to l using a set of input membership functions. They used the Mamdani inference system. They tested two models of neural network, multilayer perceptron’s (MLP) and radial basis function (RBF) networks. They proposed the Takagi-Sugeno integrated neuro-fuzzy system. The neural network learning algorithms was used to determine the parameters of the fuzzy inference system. The recognition rate was 95.64%.

In 1997, they proposed an on-line Arabic handwriting character system using fuzzy neural network as classifier. The system used a genetic algorithm to select the best combination of characters. Each character was represented by 6 feature vectors of n elements each (the n parameters Pi of the velocity profile of each stroke). The data set used for training was 2000 characters written by the one single writer [80].

In 1996, they designed a recognition system for recognizing printed Farsi/ Arabic characters with various fonts. They used Pseudo-Zernike Moments as input features which has been used for size, rotation and translation invariant. In the recognition stage, they used a combination of neural networks and fuzzy neural networks and this stage was done in two phases. In the first phase the output of these two networks were compared. In the second phase the results were corrected using 3700 character as training dataset. The Neural network had 36 inputs, 40 hidden nodes and 18 output nodes samples. Total accuracy was 99.35% [81].

In [86] they presented a new method based on structural characteristics and a fuzzy classifier for off-line recognition of handwritten Arabic characters in all their forms (beginning, end, middle and isolated). They applied three preprocessing operations on character images: thinning, contour tracing and connected components detection. Invariants pseudo-Zernike moments were used as features extracted techniques. The classification method, Fuzzy ARTMAP neural network and Five Fuzzy ARTMAP neural networks were employed; each one is designed to recognize one subset of characters. They achieved recognition process in two steps: in the first one, a clustering method affects characters to one of the five character subsets. In the second one, the pseudo-Zernike features are used by the appropriate Fuzzy ARTMAP classifier to identify the character. Training process and tests were performed on a set of character images manually extracted from the IFN/ENIT database. This system achieved 93.8% recognition rate.

In [87] they proposed a neuro-fuzzy inference engine to recognize the Farsi numeral characters. They used Mamdani inference engine on fuzzy rules with a multi-layer perceptron neural network's learning on features of the different fonts' characters which leads to more comprehensive recognition of Farsi numeral characters in the proposed system. The recognition rates of testing dataset of numeral characters are greater than 97%.

In [88] they presented an off-line Multiple Classifier System (MCS) for Arabic handwriting recognition. The MCS combined two individual recognition systems based on Fuzzy ART network used for the first time in Arabic OCR, and Radial Basis Functions. They used various feature sets based on Hu and Zernike Invariant moments. For deriving the final decision, different combining schemes are applied. The best combination ensemble has a recognition rate of 90.1 %.

3.6 Multi-Stages Classification

In [89] they proposed an offline character recognition system for isolated Arabic alphabet written by a single writer. The proposed a multiple classifier system for handwritten Arabic alphabet recognition, which has achieved an increase of about 27% in the recognition accuracy compared to a single classifier system. They followed five stages to end up with average recognition accuracy of 97% of isolated Arabic handwritten alphabet and maximum accuracy of 98.6% with Increase of about 27% from the recognition accuracy achieved by a single classifier system. However, these results were achieved for a single writer data base.

They introduced [90] an online isolated Arabic handwritten character recognition system. Feed forward back propagation neural networks are used in the classification process. Feature extraction selected were Density, Aspect Ratio and Character Alignment

Ratio Clustering character into four groups that lead to design the system as four NN with a small number of features in each decrease system complexity and increase the accuracy.

In [91] proposed a handwriting Arabic character recognition method using LeNet NN. He designed neural network with two main stages recognized character shape in the first stage using pixel matrix of 16×16 as features inputs. In the second stage, he used propagation algorithms to recognize the number of dots, position and where it is a dot or zigzag using back.

In [92] many researchers Describes a technique for the recognition of Persian handwritten isolated characters with a two-stage SVM based scheme, in the first stage, they are categorized similar shaped characters into eight groups and obtained a recognition results. In the second stage, they selected groups containing more than one similar shape characters are considered further for the final recognition, they are used features techniques based on under sampled bitmaps technique and modified chain-code direction frequencies, they computed 49 dimension features based on under sampled bitmaps. 196 dimension chain-code direction frequencies, the system achieves an accuracy rate 96.68%.

In [93] authors stated that "There are many similarities between Arabic characters in terms of structural and morphology. There are many number and position of dots that differentiate between the otherwise similar characters like (ح, ج and خ) and (ت,ن□, ب). This is the similarity between the letters that makes the Arabic character recognition difficult".

In this dissertation, neural-fuzzy with two learning algorithms model, hybrid and back-propagation to solve recognition problem of handwritten Arabic characters will be presented. Fuzzy and neural networks are used. System works by interpreting the fuzzy rules in terms of the neural network. Fuzzy sets are taken as weights, while fuzzy rules, input and output variables are taken as neurons.

3.7 Summary

This chapter has presented a general overview of the Handwritten Arabic Character Recognition (HACR) modeling concept, and a literature review on the analytic methodologies used to model the Handwritten Arabic character recognition. In the first stage, preprocessing has been presented. Although, features extraction techniques have been commonly used to model HACR. The last stage of the methodology which is classification algorithms were also discussed.

The next chapter introduces the research methodology which describes the dissertation proposed method. The solution is improved using the ANFIS with two stages of classification.

CHAPTER IV

RESEARCH METHODOLOGY

In this chapter, we focus on the methodologies of the handwriting character recognition problem. The stages of character recognition to train the data set and recognition modules will be described. The Adaptive Neural Fuzzy Inferences System Architecture and two stage classifiers approach will also be presented.

4.1 Character Recognition System Stages

A two stages-classifier system classification for Arabic handwritten isolated character is proposed. The research methodology depends on four main stages which are preprocessing, feature extraction, classification and post-processing as shown in figure 4.1. The preprocessing is utilized to remove noise and after that extracts proper image features from each character. Then these features are used for classification to identify the character through the use of the proposed neuro-fuzzy algorithm which will be applied in two classifier stages. In the last step, the algorithm will tested using a dataset.

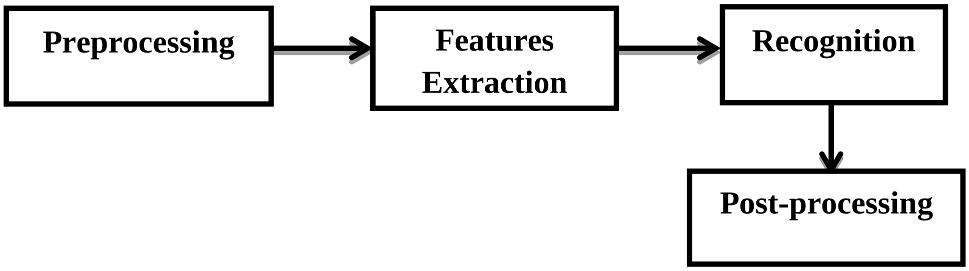
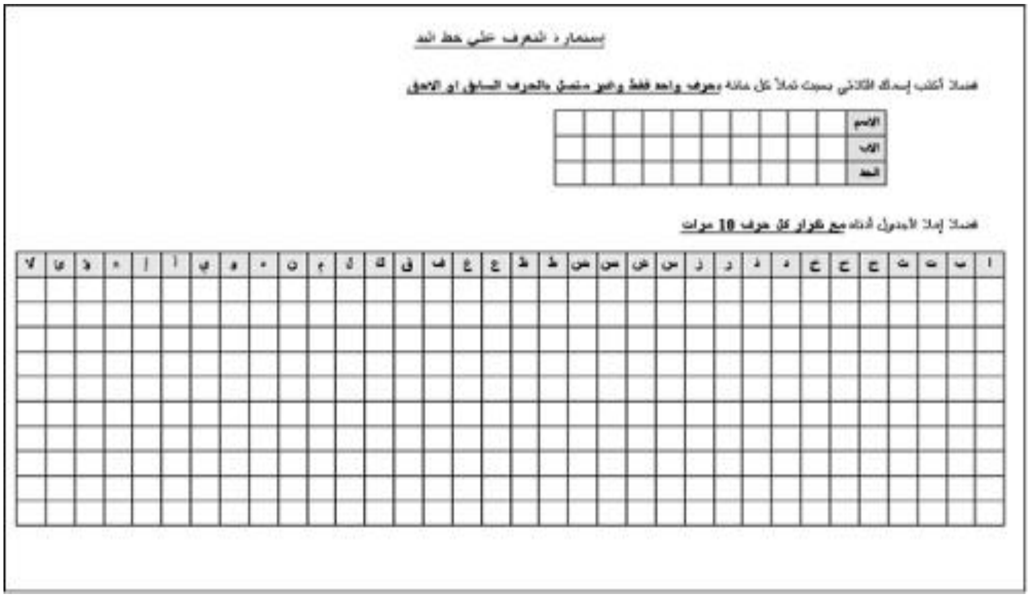


Figure 4. 1: Pattern recognition stages.

4.2 Collecting Data Set

This section describes the data set which will be used in this research. SUST-ARG dataset which stands for Sudan University for Sciences and Technology Arabic Recognition Group (SUST-ARG) is used in this research. The dataset includes one hundred and forty one forms filled by different subjects. The form as shown in figure 4.2 has been designed to collect the required handwritten letters. These forms are scanned by a scanner with accuracy of 300 dpi, saved as color images. Figure 4.3 shows the letter extraction process. This process extracts each specific letter from all forms and put it in a separate folder as a gray scale image. All letters are composed from 1410 images written by hand.



إستمارة التعرف علي خط اليد

فضلاً أكتب إسمك الثلاثي بحيث تملأ كل خانة بحرف واحد فقط وغير متصل بالحرف السابق او اللاحق

										الاسم
										الاب
										الجد

فضلاً إملأ الجدول أدناه مع تكرار كل حرف 10 مرات

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

Figure 4. 2: Illustrates the alphabet Arabic letter collection form.

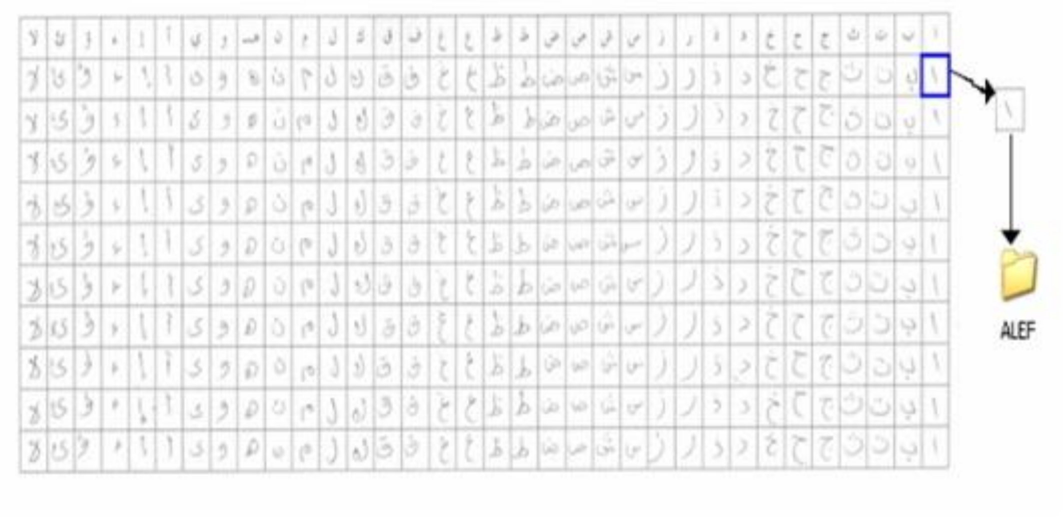


Figure 4. 3: Illustrates the letter extraction process.

4.3 The Recognition Process

This section will be presented preprocessing techniques, the techniques that are used to reduce the dimensionality of the data set as well as the feature extraction methods. Finally, it will also present an algorithm for recognizing handwritten Arabic characters that will achieve the research goals of this dissertation.

In this proposed system for recognition of handwritten isolated Arabic character, three main process are included, preprocessing, feature extraction, and classification as shown in figure 4.4. The preprocessing is utilized to remove noise and afterwards quite features are extracted from each character. Then these features are used for classification to identify the character using the proposed neuro- fuzzy algorithm.

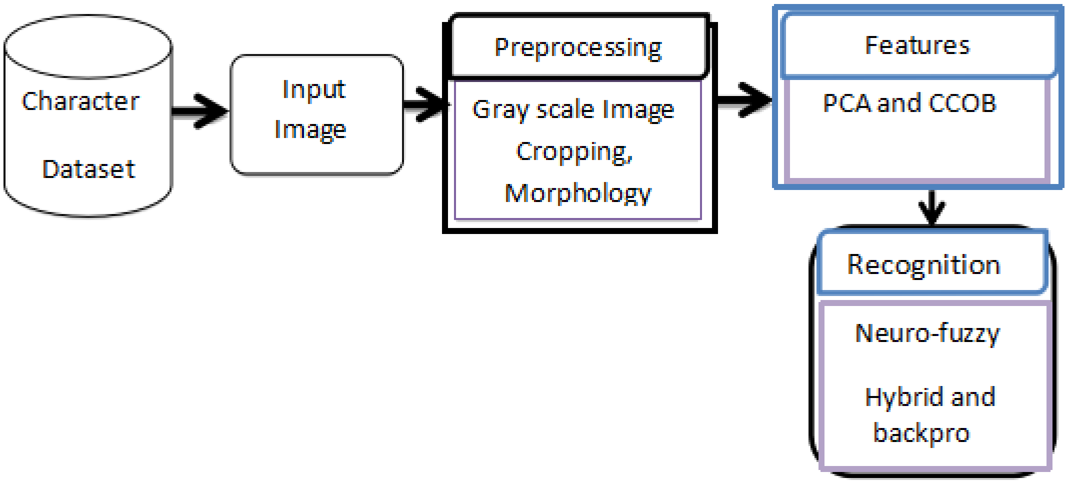


Figure 4. 4: Illustrates the recognition system steps.

4.3.1 Preprocessing

A handwriting character was sampled on A4 size paper. The characters were scanned using scanner devices at a resolution of 300dpi, then the characters were segregated according to their own character group and stored as grey scale images. Figure 4.5 shows a sample of the characters ba and kha. Cropping and resized as image preprocessing techniques were used.

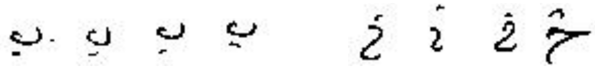


Figure 4. 5: A sample of characters (ba) and (kha) scanned and collected together.

4.3.1.1 Binarization

In the pre-processing stage the input RGB image was converted into gray scale image. Binarization is the process of converting a gray scale image (0 to 255 pixel values) into binary image (0 and 1 pixel values) by selecting a global threshold that separates the foreground from background [94]. Each pixel is compared with the threshold and if it is greater than the threshold it is made 1 or else 0. Binarization is usually reported to be performed either globally or locally. Global methods apply one intensity value to the entire image. The histogram of gray scale values of a document image typically consists of two peaks: a high peak corresponding to the white background and a smaller peak corresponding to the foreground. The task of determining the threshold gray-scale value is one of determining as optimal value in the valley between the two peaks. Local or adaptive thresholding methods apply different intensity values to different regions of the image.

4.3.1.2 Normalization

The process of changing the intensity value of the pixel to the range of [0, 1] is called normalization in image processing. The conversion of various dimension images into fixed dimensions is also called as normalization. Normalization is used in digital signal processing.

4.3.1.3 Morphological

Morphological image processing is the task of extracting image components that are useful in the representation and description of region shape. Morphology traces the outline of an object in binary image as shown in figure 4.6. Nonzero pixels belong to an object and 0 pixels constitute the background.

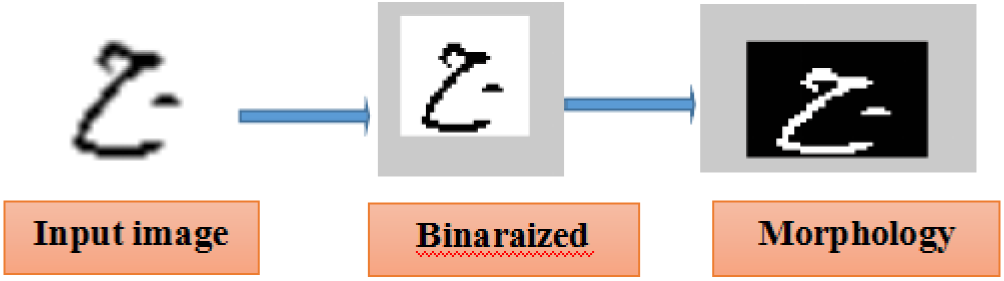


Figure 4. 6: Image morphology.

4.3.1.4 Cropping the Image

Cropping refers to the removal of the outer parts of an image to improve structure. It is removing the unimportant areas from a photographic of an image which will eliminate all the unwanted parts that are called noise. This process makes the feature more accurate and accordant.

Cropping technique steps starts first by scanning the image row by row. The last set of rows from the top and the last set of rows from the bottom with all pixels having a value of "255" is obtained. In this process cropped the top and bottom part of the image. In the second step crop the left and right side of the image. After that the image is scanned column by column. The last column from the left with all pixels having a value of "255" is obtained. The same is done to the right side of the image. Figure 4.7 showed grayscale Ta letter image after cropped and resized.

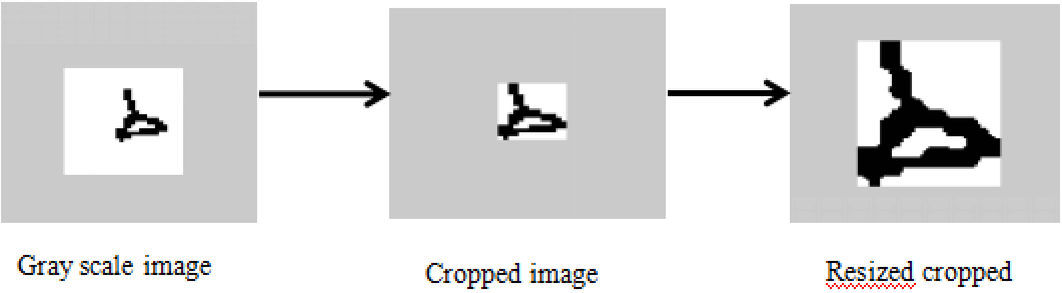


Figure 4. 7: Grayscale Tta letter image after cropped and resized.

4.3.2 Features Extraction

The purpose of feature extraction is to reduce the original data set by measuring certain properties, or features, that distinguish one input pattern from another pattern. Many characters features based on copped gray level and binarized image have been used in this dissertation.

In this approach two types of statistical features are employed. The first type of features are as follows: the area of character image is formed from the projections of the upper and lower as well as of the left and right character profiles is calculated, the center mass (xt, yt) of the character image, count the number of transitions, Outliers (Right, Left, Top and Down) and Black ink histograms is found of character image. The second sets of features uses PCA. The process of PCA starts by taking the mean of the data matrix and subtract the mean from the data matrix after that find Eigenvalues and Eigenvectors. In the last step the desire number of Eigenvectors are selected.

4.3.2.1 The First Set of Features

The statistical features used for extracting features of Arabic handwritten characters listed as following:

1. **Center of mass**

The center of mass feature f_m = (X, Y) is the relative location (relative to the height and width of the image) of the center of mass of the black ink. The center of mass of the letter jeem is shown in Figure 4.8. Given an image and a letter, where c and c' are their center of mass, respectively, the center of mass penalty used is:

$$pm = 1/(1 + d^{1/2} (c, c')) \qquad (4.1)$$

Where $d^{1/2}$ gives the square-root of the Euclidean distance (a commonly used measure). , the center of mass feature is (X/W, Y/H), where H and W are high and width.

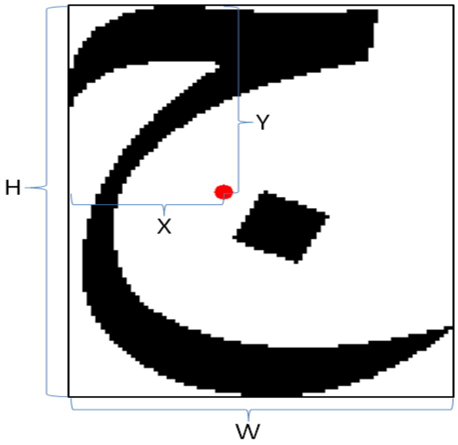


Figure 4. 8: The center of mass of the letter jeem.

2. **Cross feature**

The crosshair feature, $f_c = (X, Y)$, is the relative location (relative to the height and the width of the image) of the vertical and horizontal slices with the largest portion of black ink compared to the white background. The crosshair of the letter jeem is shown in Figure 4.9. The cross hair feature is (X/W,Y/H).

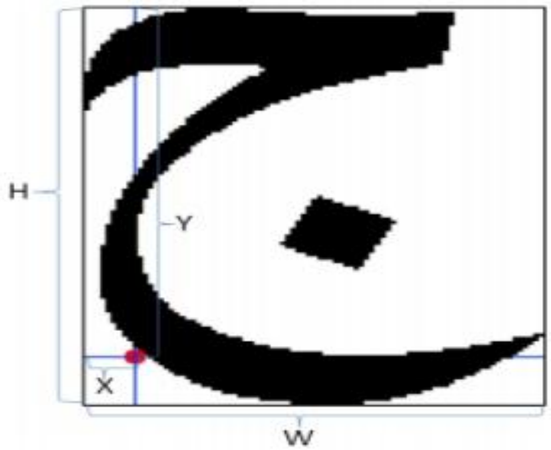


Figure 4. 9: The cross hair of the letter jeem is marked by the red dot.

3. **Outliers** (Right, Left, Top and Down) - calculate the distances of the first image pixel detected from the upper and lower boundaries of the image along the vertical lines and from the left and right boundaries along the horizontal lines. The outliers of the letter jeem is shown in Figure 4.10.

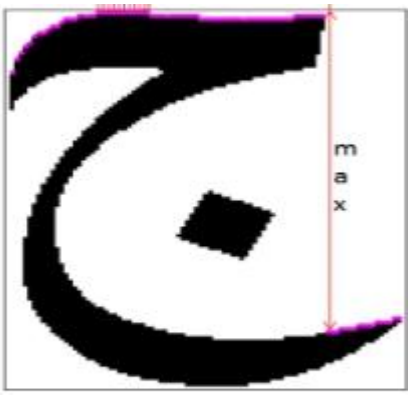


Figure 4. 10: The top outline of the letter jeem is marked by the semi-lled purple dots.

4. **Black ink histograms**

(Horizontal and Vertical) - Each image has a horizontal and vertical black ink histogram. The horizontal black ink histogram feature, fh = (h1, . . . , hH), where H is the height of the bounding box of the black ink which is calculated as follows:

a. For i = 1, ..., H, let b_i be the number of black ink pixels in row i.
b. For i = 1, ..., H, let h_i be b_i/ max{b_i}.

The vertical black ink histogram feature (fv) is calculated in a similar manner. The black ink histogram of the letter jeem is shown in Figure 4.11.

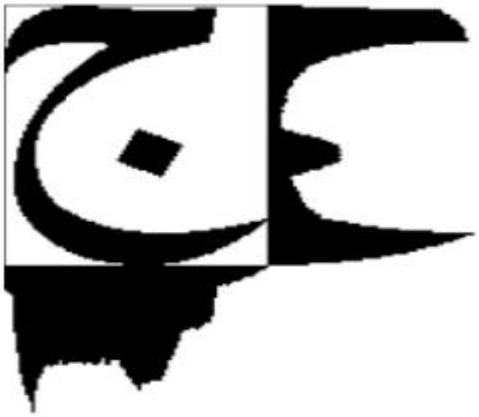


Figure 4. 11: Black ink histograms (Horizontal and Vertical).

This algorithm of features extraction is mentioned in [95] and [96]. The Outliers feature after calculated it quantized into the size of 5 values (trial and error result) for each outlier (Right, Left, Top and Down) which yields outliers feature vector containing 20 value. Similarly the Black ink histograms feature is quantized after calculation into the size of 21 values (trial and error result) for each histogram (Horizontal and Vertical) which yields Black ink histograms feature vector containing 42 values.

4.3.2.2 **Principal Components Analysis (PCA)**

PCA is a very popular technique for dimension reduction. In [97] authors have de-noised images which were given to the next process in order to calculate the score values using the PCA technique. The score values that were obtained from the PCA technique were then used by ANFIS classifier for accomplishing the training process. Based on the predefined threshold value the image under test is indicated as recognized or not recognized.

In [64] the researchers proposed five Arabic handwritten character recognition classifiers system. They used 95 feature vectors, secondary component features, main body features, skeleton features and boundary features, then they applied the PCA.

In this dissertation the PCA will be used as features extraction technique. The process of PCA starts by taking the mean of the data matrix and subtract the mean from the data matrix after that finding the Eigenvalues and Eigenvectors. The last step would select the desire number of Eigenvectors as shown in figure 4.12.

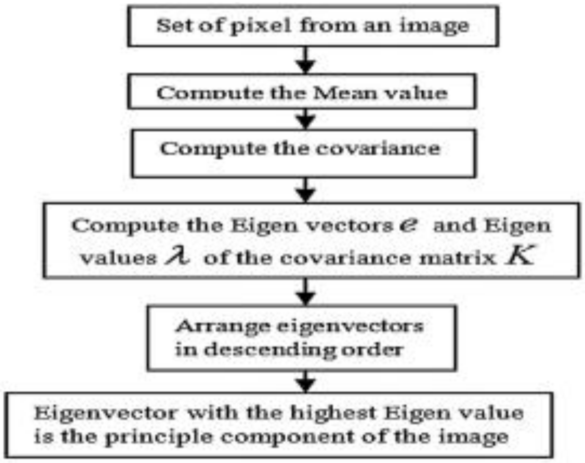

Set of pixel from an image
Compute the Mean value
Compute the covariance
Compute the Eigen vectors e and Eigen values λ of the covariance matrix K
Arrange eigenvectors in descending order
Eigenvector with the highest Eigen value is the principle component of the image

Figure 4. 12: The flow chart that illustrates the PCA.

2. PCA Algorithm

The algorithm steps are as follows:

1. Organize the image data set into column vectors

The vector size equals the image height multiplied by the image width. All the data set images must be of the same size. The result is a vector_size x database_size matrix. The matrix size (Column vectors) in the case is 7 x 5.

2. Find the empirical mean vector. Find the empirical mean along each dimension. The result is a vector_size x 1 vector, EmpiricalMean.

3. **Subtract the empirical mean vector** EmpiricalMean from each column of the data matrix ColumnVectors. Store the mean-subtracted data in a vector_size x database_size matrix, MeanSubracted.

4. Compute the eigenvectors and eigenvalues of the covariance matrix of MeanSubracted

Finding the eigenvectors and eigenvalues of the covariance matrix of MeanSubracted, which is a vector_size x vector _size matrix, was more than Matlab could handle. In order to convert these eigenvectors to the eigenvectors required to multiple the eigenvectors by the MeanSubracted matrix.

5. Sort the eigenvectors by decreasing eigenvalue.

6. Create a k dimensional subspace.

Save the first k eigenvectors as a matrix, Sub space. Eigenvector with normalized eigenvalue close to zero will not be saved even though it is in the k largest eigenvalues. The end result is a k (or less) dimensional subspace.

4.3.3 Classification

The classification process is carried out at the final stage to recognize characters. It assigns an input character to one of many pre-specified classes which are based on the extracted features. For the classification process, ANFIS (Adaptive Neural Network Fuzzy Inference System) was used in this dissertation with different algorithms.

The proposed NEURO-FUZZY topology is shown in figure 4.13. Many experiments have been conducted to find the best values of outputs (z).

The ANFIS architecture consists of fuzzification layer, inferences process, defuzzification layer, and summation as final output layer. Typical architecture of ANFIS is shown by Figure 4.13. The process flows from layer 1 to layer 5. It is started by giving a number of sets of crisp values as input to be fuzzified in layer 1, passing through inference process in layer 2 and 3 where rules applied, calculating output for each corresponding rules in layer 4 and then in layer 5 all outputs from layer 4 are summed up to get one final output. The main objective of the ANFIS is to determine the optimum values of the equivalent fuzzy inference system parameters by applying a learning algorithm using input-output data sets. The parameter optimization is done in such a way during training session that the error between the target and the actual output is minimized. Parameters are optimized by hybrid algorithm which combines the least square estimate and gradient descent method. The parameters to be optimized in ANFIS are the premise parameters which describe the shape of the membership functions, and the consequent parameters which describe the overall output of the system. The optimum parameters obtained are then used in testing session to calculate the prediction [98].

5.4 ANFIS Classifier Approach Architecture

The structure of ANFIS in this dissertation consists of three, four and five inputs and single output. The inputs represent the different character features calculated from each image. Each of the training sets forms a fuzzy inference system with different number of fuzzy rules. Each input was given three triangular membership functions and the output was represented by three nonlinear membership functions. The outputs of the fuzzy rules are condensed into one single output, representing that system output for that input image.

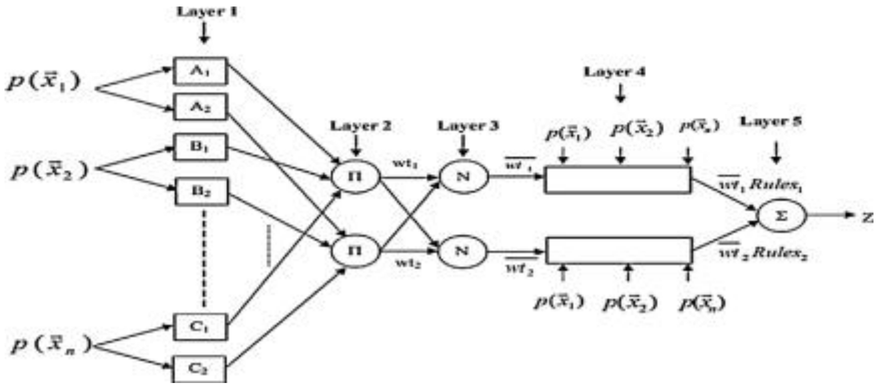
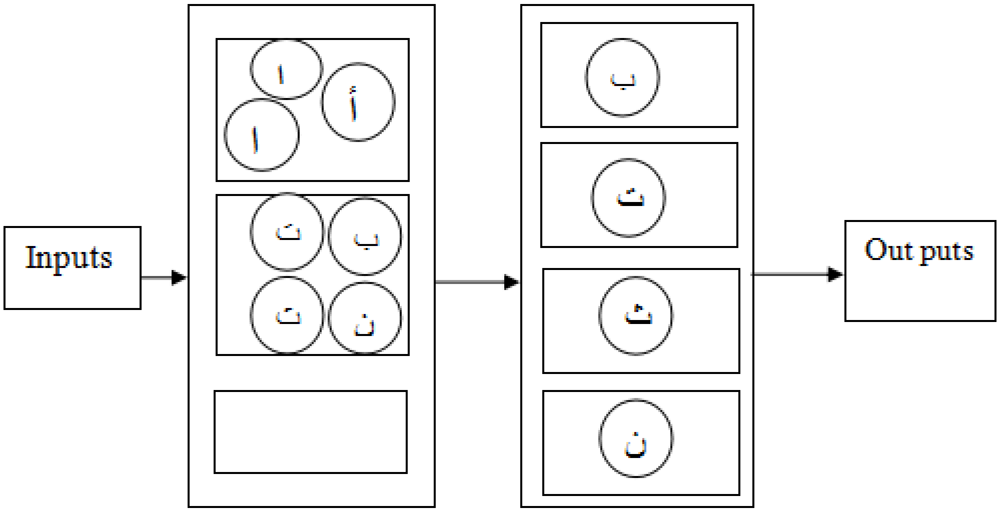


Figure 4. 13: Illustrates ANFIS architecture for Sugeno type reasoning.

4.5 Two stage classifiers approach

Two phases approach was used to deal with the issue of isolated Arabic handwritten character recognition; through combining two levels based on two classifiers, the public and the private according to the similar features between characters. First, determine the similar Arabic letters. Recognition experiments of Arabic handwritten letters data set have been conducted. Each letter represents a different class with neural fuzzy classifier. After the recognition result, conducting pro-processing work with confusion matrix, this determines the similarity among the letters. The next step is to collect the similar letters in one class as shown in figure 4.14(a), with equal numbers for each letter. A classifier for each class of the similar letters would be designed so that each character will be in one class as shown in figure 4.14(b). When running a full system, first determine the similar characters that would belong to on class, then in the second stage, determine the specific letter. All classifiers will be designed with neuro-fuzzy algorithm as shown in figure 4.15.



(a) (b)

Figure 4.14: Illustrate new proposed method.

(a) Illustrate data set of Arabic character groups, each class includes similar characters.

(b) Illustrates data set of the similar Arabic characters, each class includes one character.

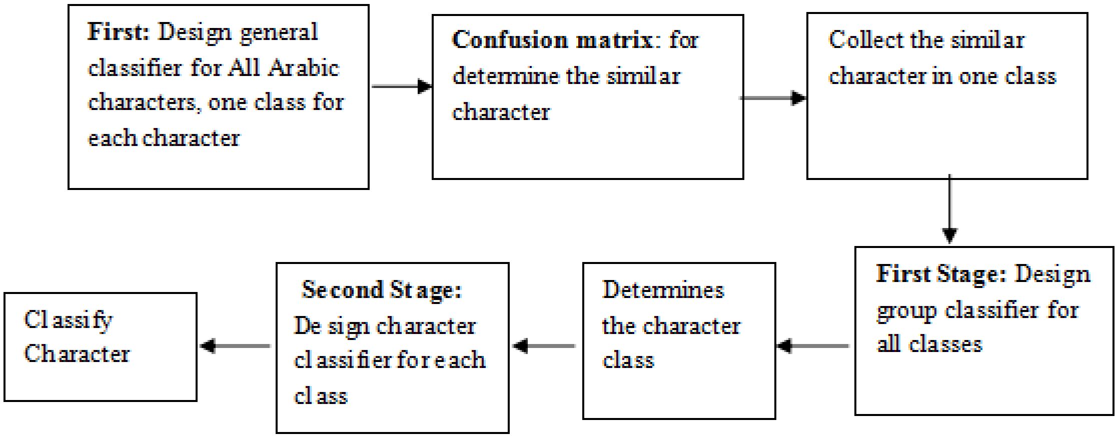

First: Design general classifier for All Arabic characters, one class for each character
Confusion matrix: for determine the similar character
Collect the similar character in one class
First Stage: Design group classifier for all classes
Determines the character class
Second Stage: Design character classifier for each class
Classify Character

Figure 4. 15: A new proposed approach with two stages of classifiers.

5.6 Summary

This chapter introduces the research methodology. It gives review of Character Recognition System Stages. Proposed a two stages-classifier system for Isolated Arabic handwritten character through combining two levels based on two classifiers (the group and the character classifiers) according to the similar features among characters. The chapter presents the design and the research stage process that contains four phases. The

SUST-ARG data set is described; preprocessing techniques used are described as well as the features extraction.

Next chapter presents the neural network system that was used to determine the similarities between Arabic characters by using confusion matrix. The simulations are done using MATLAB software package.

CHAPTER V

SIMILAR CHARACTERS EXPERIMENTS WITH ANN CLASSIFIER

This chapter described the topology of Artificial Neural Network (ANN) that was used for determining the characters that have similar structure. Section 1, describes introduction, section 2 the first experiments with different features. Section 3, describes second experiments. Section 4, presented confusion matrixes for two experiments. Section 5, explains the similarities between characters by applied the confusion matrix.

5.1 Introduction

The experiments were conducted on the isolated Arabic handwriting letters which are part of the SUST-ARGG data set. Back-propagation neural network algorithm was used to find best values for the number of input features and hidden layer units. The proposed network topology is shown in figure 5.1. Two size handwritten characters data set were used. The first size is 10200 images divided into 200 samples for each class for training and 100 samples for testing. The second size used was 23800 images divided into 700 samples for training and 200 for testing. The handwritten character images are taken from the SUST-ARGG dataset. These images are binarized and cropped in preprocessing stage and passed to the features techniques and extracted different number of features.

5.2 First Experiments

For the experiments, 66% of the data set was selected for training and 33% were selected for testing. Principle Component Analysis (PCA) is also used on set of all features to reduce the dimension from 35 to 30, 20, 15 and 10. 200 samples of the data set were selected for training and 100 for testing in conducting many experiments. The result of these experiments is shown in table 5.1.

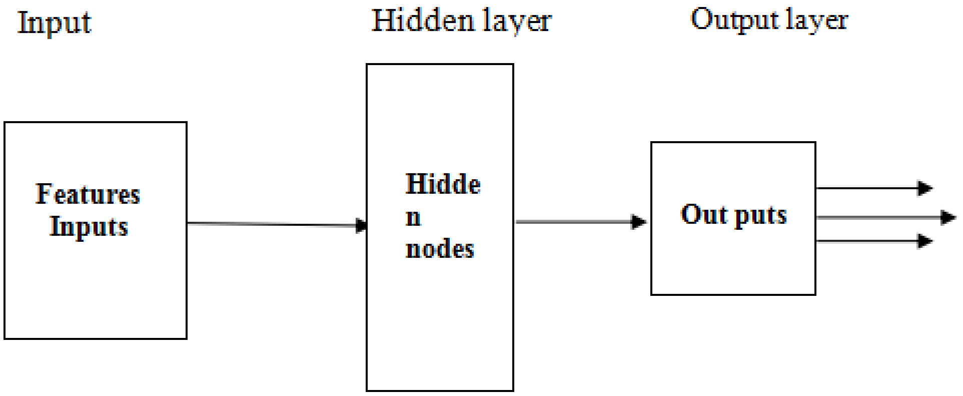


Figure 5. 1: Network topology.

The best ANN for recognition accuracy is the 150 nodes with 25 features as inputs and 89 epochs as shown in figure 5.2. On training, it recorded an accuracy of 84.6%. On test it had an accuracy of 65.5%.

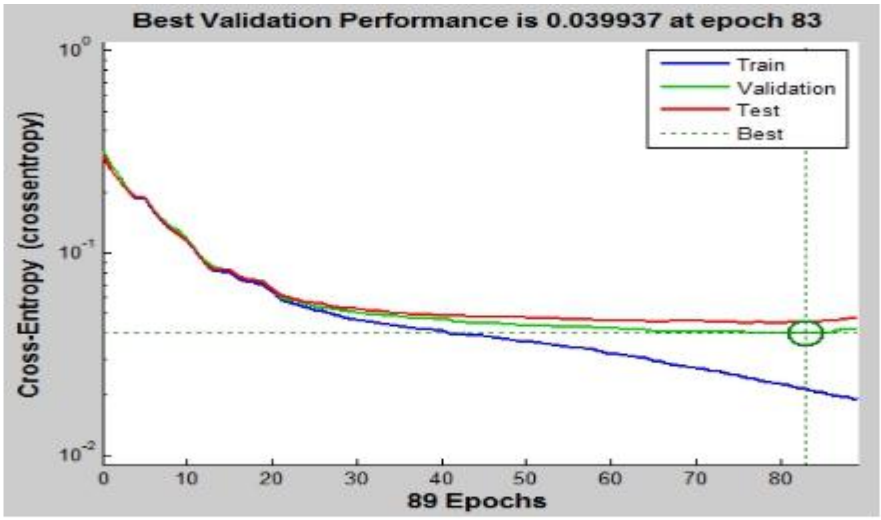


Figure 5. 2: 150 nodes with 25 features.

Table 5. 1: Accuracy rate for character data set with different features.

Feature	**Hidden nodes**	**Time seconds**	**Train accuracy %**	**Test accuracy %**
35	100	6	82	57.2941
	150	9	87.6863	61.2353
	200	10	87.2353	61.5882
30	100	5	76.4902	59.2941
	150	10	84.8627	60.2941
	200	10	83.451	60.7647
25	100	9	80.4706	65.4118
	200	17	83.1176	64.5588
	150	**16**	**84.5588**	**65.4706**
20	100	5	76.8235	58.4118
	150	8	78	59.3529
	200	7	73.3137	58
10	100	8	75.2745	58.5882
	150	10	73.3529	57.9412
	200	15	76.8824	58.4706

5.3 The Second Set of Experiments

For the experiments, 78% of the data set was selected for training and 22% for testing. Principle Component Analysis (PCA) is also used on set of all features to reduce the dimension from 900 to 35, 30, 25, 20 and 10.

700 samples were selected for each character as training data set and 200 samples for testing data set for conducting many experiments. The result of these experiments is shown in table 5.2.

The best ANN for recognition accuracy is the 100 node with 100 features as inputs. On training, it recorded an accuracy of 58.6%. On test it had an accuracy of 48.06%. Table 5.9 and table 5.10 show the confusion matrices for the testing sets.

Table 5. 2: Accuracy rate for character data set with different features.

Feature	Hidden units	Time seconds	Train accuracy %	Test accuracy %
100	**100**	**33**	**58.6933**	**48.0588**
	300	131	45.1008	28.5441
	600	160	65.084	45.9706
200	100	54	57.9664	40.5147
	300	134	61.0924	38.9559
	600	194	48.8151	35.7941

5.4 Confusion Matrix

A confusion matrix is a table that contains information about actual and predicted classifications that are done by a classification system (or "classifier") on a set of data to know the true values and to enhance the recognition accuracy of the AHC system [99]. Sections 5.4.1 and 5.4.2 below explain the confusion matrix for two experiments. Table 5.3 shows the sort of characters in the confusion matrix.

5.4.1 Confusion Matrix for the First Set of Experiments

This confusion matrix is generated from the data prepared in section 5.2 which is used in order to train and test the 34 class of Arabic handwritten character for the high recognition rate as shown in table 5.1. Table 5.4 shows the confusion matrix for testing sets.

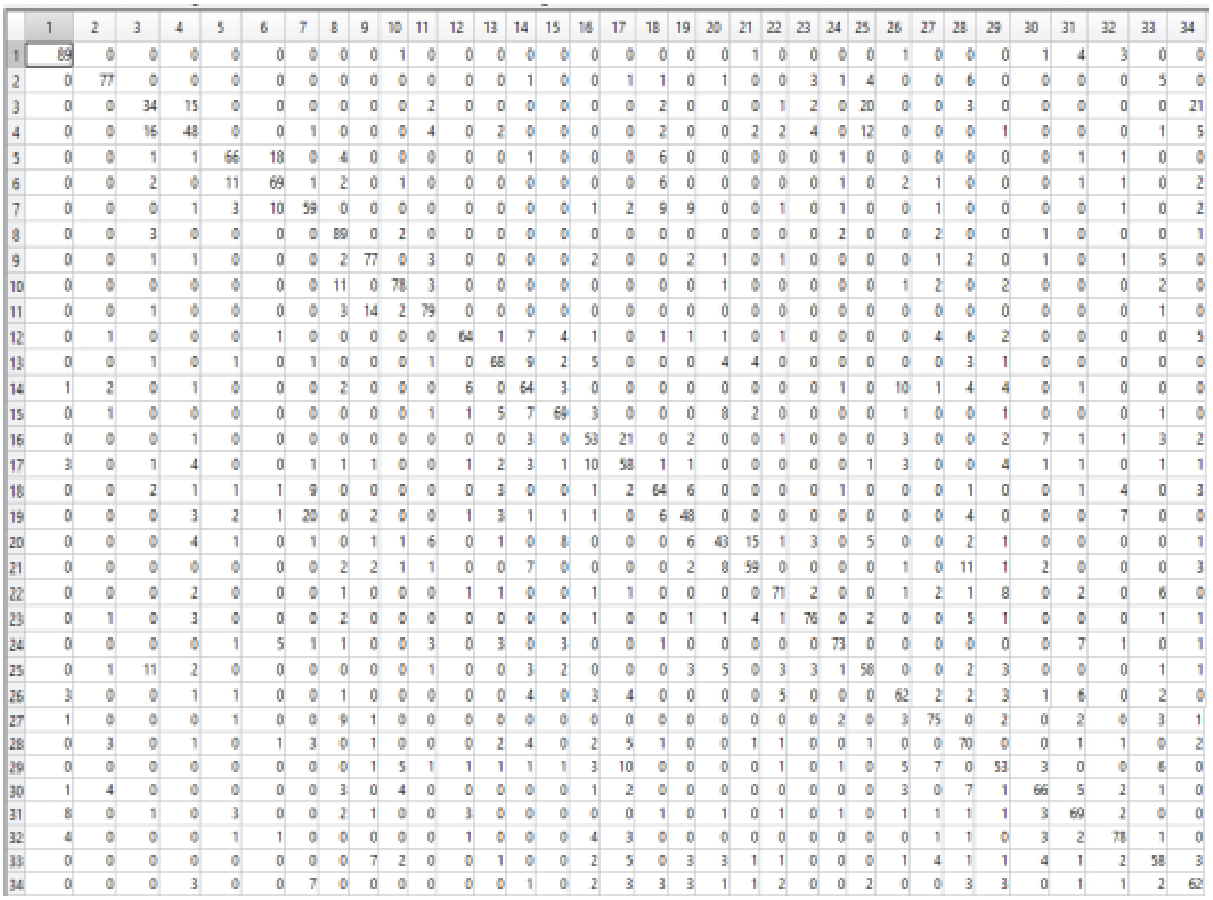
The confusion matrix as shown in table 5.3 reflects the fact that the error occurred between similar-shaped letters. For example, out of 100 samples for the letter “ت” only 34 samples are correctly recognized. Regarding many of the number of characters out of 66 were recognized as other characters, as shown in table 5.5. For example, 20% of the character "ت" were recognized as "ن" and 30% of the character "ج" were recognized as "ح" which is another indicative instance for this similar shaped letters confusion as shown in table 5.5 and table 5.7.

Table 5. 3: Illustrates Character numbers in the confusion matrix.

Characters	N	characters	N
ع	18	ا	1
غ	19	ب	2
ف	20	ت	3
ق	21	ث	4
ك	22	ج	5
ل	23	ح	6
م	24	خ	7
ن	25	د	8
هـ	26	ذ	9
و	27	ر	10
ي	28	ز	11
لا	29	س	12
ء	30	ش	13
أ	31	ص	14
إ	32	ض	15
ؤ	33	ط	16
ئ	34	ظ	17

The low recognition rate for the letter "ت" is due to the strong similarity between character "ث" and character "ن". The high reliability recognition rate for the letter " ا " is due to the clear difference between it and all the characters except for some writers in the way they draw the letters "أ" and "إ" as shown in table 5.6.

Table 5. 4: Testing 100 images data set confusion matrix.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	89	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	4	3	0	0
2	0	77	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	3	1	4	0	0	6	0	0	0	0	5	0
3	0	0	34	15	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	1	2	0	20	0	0	3	0	0	0	0	0	21
4	0	0	16	48	0	0	1	0	0	0	4	0	2	0	0	0	0	2	0	0	2	2	4	0	12	0	0	0	1	0	0	0	1	5
5	0	0	1	1	66	18	0	4	0	0	0	0	0	1	0	0	0	6	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
6	0	0	2	0	11	69	1	2	0	1	0	0	0	0	0	0	0	6	0	0	0	0	0	1	0	2	1	0	0	0	1	1	0	2
7	0	0	0	1	3	10	59	0	0	0	0	0	0	0	0	1	2	9	9	0	0	1	0	1	0	0	1	0	0	0	0	1	0	2
8	0	0	3	0	0	0	0	89	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0	0	1	0	0	0	1
9	0	0	1	1	0	0	0	2	77	0	3	0	0	0	0	2	0	0	2	1	0	1	0	0	0	0	1	2	0	1	0	1	5	0
10	0	0	0	0	0	0	0	11	0	78	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	2	0	2	0	0	0	2	0
11	0	0	1	0	0	0	0	3	14	2	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
12	0	1	0	0	0	1	0	0	0	0	0	64	1	7	4	1	0	1	1	1	0	1	0	0	0	0	4	6	2	0	0	0	0	5
13	0	0	1	0	1	0	1	0	0	0	1	0	68	9	2	5	0	0	0	4	4	0	0	0	0	0	0	3	1	0	0	0	0	0
14	1	2	0	1	0	0	0	2	0	0	0	6	0	64	3	0	0	0	0	0	0	0	0	1	0	10	1	4	4	0	1	0	0	0
15	0	1	0	0	0	0	0	0	0	0	1	1	5	7	69	3	0	0	0	8	2	0	0	0	0	1	0	0	1	0	0	0	1	0
16	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0	53	21	0	2	0	0	1	0	0	0	3	0	0	2	7	1	1	3	2
17	3	0	1	4	0	0	1	1	1	0	0	1	2	3	1	10	58	1	1	0	0	0	0	0	1	3	0	0	4	1	1	0	1	1
18	0	0	2	1	1	1	9	0	0	0	0	0	3	0	0	1	2	64	6	0	0	0	0	1	0	0	0	1	0	0	1	4	0	3
19	0	0	0	3	2	1	20	0	2	0	0	1	3	1	1	1	0	6	48	0	0	0	0	0	0	0	0	4	0	0	0	7	0	0
20	0	0	0	4	1	0	1	0	1	1	6	0	1	0	8	0	0	0	6	43	15	1	3	0	5	0	0	2	1	0	0	0	0	1
21	0	0	0	0	0	0	0	2	2	1	1	0	0	7	0	0	0	0	2	8	59	0	0	0	0	1	0	11	1	2	0	0	0	3
22	0	0	0	2	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0	0	71	2	0	0	1	2	1	8	0	2	0	6	0
23	0	1	0	3	0	0	0	2	0	0	0	0	0	0	0	1	0	0	1	1	4	1	76	0	2	0	0	5	1	0	0	0	1	1
24	0	0	0	0	1	5	1	1	0	0	3	0	3	0	3	0	0	1	0	0	0	0	0	73	0	0	0	0	0	0	7	1	0	1
25	0	1	11	2	0	0	0	0	0	0	1	0	0	3	2	0	0	0	3	5	0	3	3	1	58	0	0	2	3	0	0	0	1	1
26	3	0	0	1	1	0	0	1	0	0	0	0	0	4	0	3	4	0	0	0	0	5	0	0	0	62	2	2	3	1	6	0	2	0
27	1	0	0	0	1	0	0	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3	75	0	2	0	2	0	3	1
28	0	3	0	1	0	1	3	0	1	0	0	0	2	4	0	2	5	1	0	0	1	1	0	0	1	0	0	70	0	0	1	1	0	2
29	0	0	0	0	0	0	0	0	1	5	1	1	1	1	1	3	10	0	0	0	0	1	0	1	0	5	7	0	53	3	0	0	6	0
30	1	4	0	0	0	0	0	3	0	4	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	3	0	7	1	66	5	2	1	0
31	8	0	1	0	3	0	0	2	1	0	0	3	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	3	69	2	0	0
32	4	0	0	0	1	1	0	0	0	0	0	1	0	0	0	4	3	0	0	0	0	0	0	0	0	0	1	1	0	3	2	78	1	0
33	0	0	0	0	0	0	0	0	7	2	0	0	1	0	0	2	5	0	3	3	1	1	0	0	0	1	4	1	1	4	1	2	58	3
34	0	0	0	3	0	0	7	0	0	0	0	0	0	1	0	2	3	3	3	1	1	2	0	0	2	0	0	3	3	0	1	1	2	62

Table 5. 5: Confusion matrix for many characters.

ن	ث	ت	ب	
4	0	0	**77**	ب
20	15	**34**	0	ت
12	**48**	16	0	ث
58	2	11	1	ن

Table 5. 6: Confusion matrix for group one characters.

إ	أ	ا	
3	4	**89**	ا
2	**79**	8	أ
78	3	4	إ

Table 5. 7: Confusion matrix for group three of characters.

خ	ج	ح	
9	18	**65**	ح
15	**66**	30	ج
52	19	10	خ

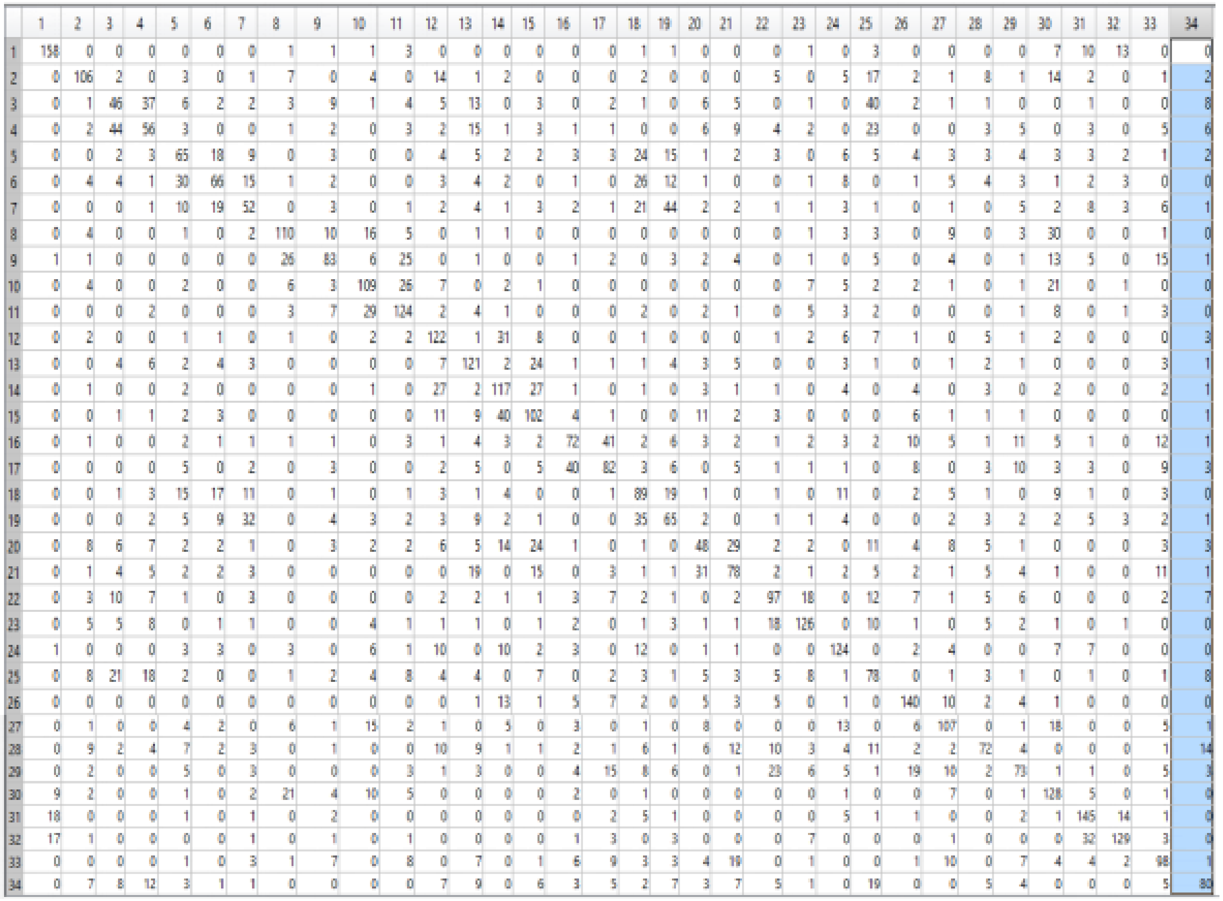
Tables 5.5, 5.6 and 5.7 show part of the generated confusion matrix which includes the ANN classifier for the first set of experiments.

5.4.2 Confusion Matrix for the Second Set of Experiments

In the second set of experiments, after a confusion matrix was designed as shown in table 5.8 several observations were described. For example, out of 200 samples to the letter “ث” only 56 samples are correctly recognized. About many of the number of characters out of 144 were recognized as other characters, as shown in table 5.10. 22% of the character “ث”, were recognized as “ت” and 15% of the character “ج”, were recognized as “ح” which is another indicative instance for this similar shaped letters confusion as shown in tables 5.9, 5.10, 5.11 and 5.12.

17% of the character “غ”, were recognized as “ع”, 16% of the character “ق”, were recognized as “ـف” as well as the vice versa 15% of character “ـف”, were recognized as”ق”. 20% of the character “ـط”, were recognized as “ـظ” and ideal of character “ـظ”, were recognized as”ـط”, and so on.

Table 5. 8: Confusion matrix for 200 images testing data set.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	158	0	0	0	0	0	0	1	1	1	3	0	0	0	0	0	0	1	1	0	0	0	1	0	3	0	0	0	0	7	10	13	0	0
2	0	106	2	0	3	0	1	7	0	4	0	14	1	2	0	0	0	2	0	0	0	5	0	5	17	2	1	8	1	14	2	0	1	2
3	0	1	46	37	6	2	2	3	9	1	4	5	13	0	3	0	2	1	0	6	5	0	1	0	40	2	1	1	0	0	1	0	0	8
4	0	2	44	56	3	0	0	1	2	0	3	2	15	1	3	1	1	0	0	6	9	4	2	0	23	0	0	3	5	0	3	0	5	6
5	0	0	2	3	65	18	9	0	3	0	0	4	5	2	2	3	3	24	15	1	2	3	0	6	5	4	3	3	4	3	3	2	1	2
6	0	4	4	1	30	66	15	1	2	0	0	3	4	2	0	1	0	26	12	1	0	0	1	8	0	1	5	4	3	1	2	3	0	0
7	0	0	0	1	10	19	52	0	3	0	1	2	4	1	3	2	1	21	44	2	2	1	1	3	1	0	1	0	5	2	8	3	6	1
8	0	4	0	0	1	0	2	110	10	16	5	0	1	1	0	0	0	0	0	0	0	0	1	3	3	0	9	0	3	30	0	0	1	0
9	1	1	0	0	0	0	0	26	83	6	25	0	1	0	0	1	2	0	3	2	4	0	1	0	5	0	4	0	1	13	5	0	15	1
10	0	4	0	0	2	0	0	6	3	109	26	7	0	2	1	0	0	0	0	0	0	0	7	5	2	2	1	0	1	21	0	1	0	0
11	0	0	0	2	0	0	0	3	7	29	124	2	4	1	0	0	0	2	0	2	1	0	5	3	2	0	0	0	1	8	0	1	3	0
12	0	2	0	0	1	1	0	1	0	2	2	122	1	31	8	0	0	1	0	0	0	1	2	6	7	1	0	5	1	2	0	0	0	3
13	0	0	4	6	2	4	3	0	0	0	0	7	121	2	24	1	1	1	4	3	5	0	0	3	1	0	1	2	1	0	0	0	3	1
14	0	1	0	0	2	0	0	0	0	1	0	27	2	117	27	1	0	1	0	3	1	1	0	4	0	4	0	3	0	2	0	0	2	1
15	0	0	1	1	2	3	0	0	0	0	0	11	9	40	102	4	1	0	0	11	2	3	0	0	0	6	1	1	1	0	0	0	0	1
16	0	1	0	0	2	1	1	1	1	0	3	1	4	3	2	72	41	2	6	3	2	1	2	3	2	10	5	1	11	5	1	0	12	1
17	0	0	0	0	5	0	2	0	3	0	0	2	5	0	5	40	82	3	6	0	5	1	1	1	0	8	0	3	10	3	3	0	9	3
18	0	0	1	3	15	17	11	0	1	0	1	3	1	4	0	0	1	89	19	1	0	1	0	11	0	2	5	1	0	9	1	0	3	0
19	0	0	0	2	5	9	32	0	4	3	2	3	9	2	1	0	0	35	65	2	0	1	1	4	0	0	2	3	2	2	5	3	2	1
20	0	8	6	7	2	2	1	0	3	2	2	6	5	14	24	1	0	1	0	48	29	2	2	0	11	4	8	5	1	0	0	0	3	3
21	0	1	4	5	2	2	3	0	0	0	0	0	19	0	15	0	3	1	1	31	78	2	1	2	5	2	1	5	4	1	0	0	11	1
22	0	3	10	7	1	0	3	0	0	0	0	2	2	1	1	3	7	2	1	0	2	97	18	0	12	7	1	5	6	0	0	0	2	7
23	0	5	5	8	0	1	1	0	0	4	1	1	1	0	1	2	0	1	3	1	1	18	126	0	10	1	0	5	2	1	0	1	0	0
24	1	0	0	0	3	3	0	3	0	6	1	10	0	10	2	3	0	12	0	1	1	0	0	124	0	2	4	0	0	7	7	0	0	0
25	0	8	21	18	2	0	0	1	2	4	8	4	4	0	7	0	2	3	1	5	3	5	8	1	78	0	1	3	1	0	1	0	1	8
26	0	0	0	0	0	0	0	0	0	0	0	0	1	13	1	5	7	2	0	5	3	5	0	1	0	140	10	2	4	1	0	0	0	0
27	0	1	0	0	4	2	0	6	1	15	2	1	0	5	0	3	0	1	0	8	0	0	0	13	0	6	107	0	1	18	0	0	5	1
28	0	9	2	4	7	2	3	0	1	0	0	10	9	1	1	2	1	6	1	6	12	10	3	4	11	2	2	72	4	0	0	0	1	14
29	0	2	0	0	5	0	3	0	0	0	3	1	3	0	0	4	15	8	6	0	1	23	6	5	1	19	10	2	73	1	1	0	5	3
30	9	2	0	0	1	0	2	21	4	10	5	0	0	0	0	2	0	1	0	0	0	0	0	1	0	0	7	0	1	128	5	0	1	0
31	18	0	0	0	1	0	1	0	2	0	0	0	0	0	0	0	2	5	1	0	0	0	0	5	1	1	0	0	2	1	145	14	1	0
32	17	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	3	0	3	0	0	0	7	0	0	0	1	0	0	0	32	129	3	0
33	0	0	0	0	1	0	3	1	7	0	8	0	7	0	1	6	9	3	3	4	19	0	1	0	0	1	10	0	7	4	4	2	98	1
34	0	7	8	12	3	1	1	0	0	0	0	7	9	0	6	3	5	2	7	3	7	5	1	0	19	0	0	5	4	0	0	0	5	80

Table 5. 9: Confusion matrix for group on in second experiment.

إ	أ	ا	
13	10	**158**	ا
14	**145**	18	أ
129	32	17	إ

Table 5. 10: Confusion matrix for group 2.

ن	ث	ت	ب	
17	0	2	**106**	ب
40	37	**46**	1	ت
23	**56**	44	2	ث
78	18	21	8	ن

Table 5. 11: Confusion matrix for group 3.

خ	ج	ح	
9	18	**65**	ح
15	**66**	30	ج
52	19	10	خ

Table 5. 12: Confusion matrix for many characters.

ء	ز	ر	ذ	د	
30	5	16	10	**110**	د
13	25	6	**83**	26	ذ
21	26	**109**	3	6	ر
8	**124**	29	7	3	ز
128	**5**	10	4	21	ء

Tables 5.9 and 5.10, 5.11 and 5.12 show part of the generated confusion matrix which includes the ANN classifier for the second set of experiments.

The low recognition rate for the letter “ت” is due to the strong similarity between character “ث” and character “ن” as shown in table 5.10. The high reliability recognition rate for the letter “ ا “ is due to the difference between it and all the characters except for some writers in which they draw the letters “أ” and “إ” as shown in table 5.6 and table 5.9.

Table 5.12 shows that the “د” letter read 30 times as a “ء” letter because many writers draw the dal letter “د” like the “Hamza” letter “ء”.

5.4 The Similarities

The recognition process was conducted on 34 numbers of classes. Those contains handwritten Arabic character are more difficult because there are numbers of classes with high similarity between them, especially for the similar structural characters. The confusion matrix tables 5.4 and 5.9 show the Arabic characters that have been miss-classified. The first and second set of experiments in section 5.3 has shown the high confusion between similar letters. To tackle these problems, the dissertation will present a proposal for a two stage of classifier method, which will be discussed in chapter 6. The researcher suggested that similar characters are distributed into two types of groups; the first type is composed of 15 classes and the second into 10 classes as show in table 5.13.

Table 5. 13: Characters has overlapped.

Group number	Ten Character Group	Fifteen Character Group
1	م ا إ أ	ا إ أ
2	ي ئ ب ت ث ن	ب ت ث ن
3	ج ح خ ع غ	ج ح خ
4	د ذ ر ز ء	د ذ ر ز ء
5	ش س ص ض ف ق	س ص ض
6	لا ط ظ	ش
7	و	ط ظ
8	ؤ	ع غ
9	ل ك	ف ق ؤ
10	هـ	ل ك
11		م
12		هـ
13		و
14		ئ ي
15		لا

5.5 Summary

In this chapter ANN model is used to design the classifiers and implementing a back-propagation neural network algorithm to training and testing data sets. We conducted two experiments on the SUST-ARG data set with different sizes. We present confusion matrix for highest accuracy recognition rate and we determined those characters has overlapped.

In the next chapter, the two stages proposed method, used the adaptive neuro-fuzzy inference system (ANFIS) to solution the problem.

CHAPTER VI

APPLICATION IN ARABIC ISOLATED HANDWRITTEN CHARACTER RECOGNITION SYSTEM EXPERIMENTS WITH ANFIS CLASSIFIER

This chapter is organized as follows: section 1 describes the data set, section 2 describes the two stage methods, section 3 describes the experiments with the first set of features, section 4 describes the experiments with the second set of features, and section 5 describes the accuracy rates for classes for single classifier, group classifier and character classifier. Section 6, describes the experiments with hybrid features. Section 7 describes the addition experiments with different grouping of data set. Section 8 discussed time and space analysis. Section 9, presents a comparative study, and section 10 presents discussion.

6.1 Introduction

Figure 6.1 shows the new proposed Arabic character recognition system. Based on experiments conducted on handwritten Arabic data set with neural network classifier and its results which showed significant similarity between some letters, the data set was divided into three different levels as shown in figure 6.2. These levels vary in size and number of classes. The characters that were overlapping as seen in the confusion matrix were grouped together. The first level contains all data and every character in a single class. The second level included data sets in group’s class where each group contains the similar characters in one class. The letters are grouped in a single class based on the noise matrix were every character interferes with the other in more than 20% of the sample. The third level for groups of similar characters so that each group contains similar characters and each character in a different class. This third level has a number of classifiers equal to a number of groups in the second level. These levels are described below as follows:

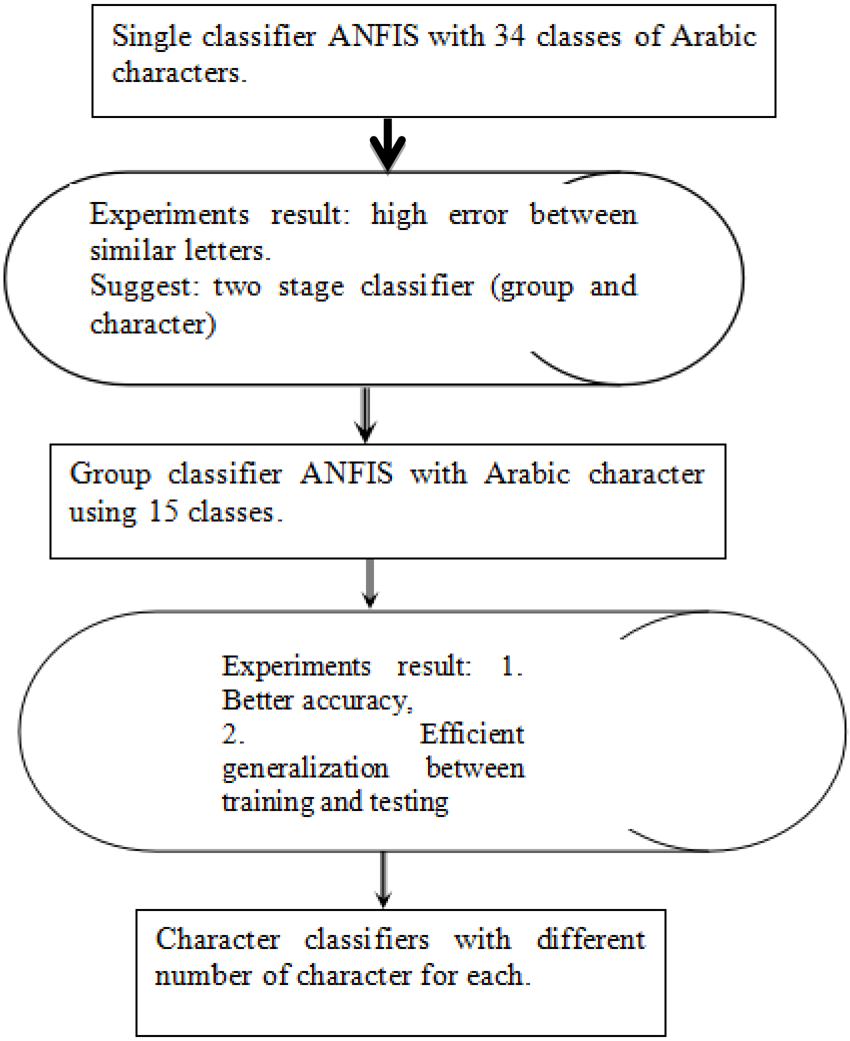

Single classifier ANFIS with 34 classes of Arabic characters.
Experiments result: high error between similar letters.
Suggest: two stage classifier (group and character)
Group classifier ANFIS with Arabic character using 15 classes.
Experiments result: 1. Better accuracy,
2. Efficient generalization between training and testing
Character classifiers with different number of character for each.

Figure 6. 1: The proposed method steps.

Step1. Features are extracted by feature extraction techniques.

Step2. Start training with a single classifier (SC), each class consists of only one character.

Step3. The highest recognition rate (HRR) is calculated to compare the result with proposed system.

Step4. The confusion matrix is created from (HRR) result.

Step5. The highest similarity between characters is determined.

Step6. Classes with high misclassification rates are merged to gather as a group (G).

Step7. Find the number of groups (N of G). Each group is made of 1 to 5 classes (N of C).

Step8. Start training with group's classifier (GC).

Step9. The highest recognition rate (HRR) for group's classifier is obtained.

Step10. Every character in the group is a separate class.

Step11. Construct character classifier (CC) for every group Gi.

Step12. Start training with characters classifier.

Step13. The highest recognition rate (HRR) for each character classifier is obtained.

In the experiments, 75% of the data set has been selected for training and 25% for testing for all levels as shown in table 6. 1.

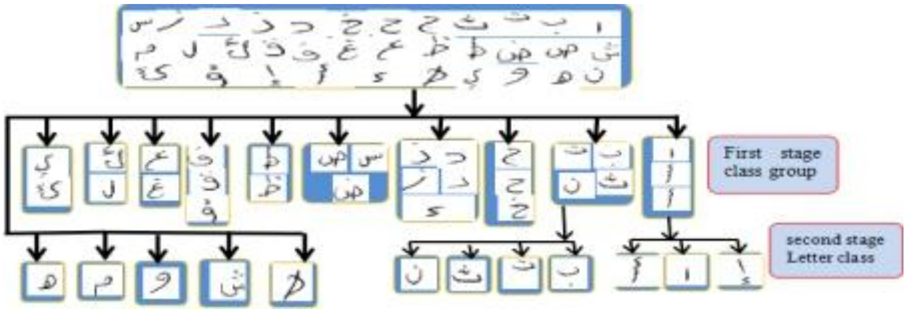


Figure 6. 2: The proposed two-stage Arabic Handwritten character recognition system.

Table 6. 1: Data set types and sizes for training and testing.

Dataset Level	Dataset size	Training dataset	Testing dataset
All dataset with 34 characters in 34 class	6800	5100	1700
Grouping dataset into 15 different classes	3000	2250	750
Similar characters (1 to 5) class for each character	each character with 200 image	150*class number	50*class number

6.2 Two Stages Method

We used a two stage classifier. The first stage is based on features extracted from groups of similar characters. These groups are determined by the ANN classifier as show in chapter 5. The second stage has sub-classifiers. A sub-classifier is a multiple adaptive neuro-fuzzy classifier to recognize the characters of only one group. In the second stage, different examples from the similar character class may fall into different groups obtained during the first stage. The group may consist of one or more character classes. In the experiments we used ANFIS classifier with two learn algorithms, hybrid and back-propagation. Figure 6.1 shows the proposed method steps.

In the experiments, have been employed two types of statistical features. The first type of features (CCOB) are the Center mass (xt, yt) of the character image, Cross hair count the number of transitions, Outliers (Right, Left, Top and Down) and Black ink histograms. The second sets of features used PCA to select the desired number of eigenvectors and eigenvalues from resize cropped images.

6.3 Experiments with First Set of Features

The handwritten character images are taken from the SUST-ARGG dataset. These images are both binarized and morphologicalized, second are passed to the feature extracted techniques and extracted four (CCOB) features.

Four features were used as inputs to the system, three numbers of triangular membership functions were used and single output as shown in figure 6.3 with 1000 epochs.

Analyzing results show the improvement at the first stage were the recognition accuracy is 96.1% for isolated Arabic handwritten character with testing data set in grouped classifiers with an increase of about 5.6% from the recognition accuracy achieved by a single classifier system as shown in table 6.2.

Figure 6.4 shows the accuracy rate results for testing data set with two learning algorithm, hybrid and back-propagation which were used by the ANFIS classifier for training the data set.

Table 6. 2: Accuracy rate for a data set with first set of features.

Dataset type	**Training accuracy %**		**Testing accuracy%**	
	Hybrid	Backpropagation	Hybrid	Backpropagation
All data set (single classifier)	90.7	90.3	90.5	90.1
Grouping data set (group classifier)	96.12	95.8	96.1	95.02
Similar data set (character classifier)	99.7	99.3	99.3	99.1

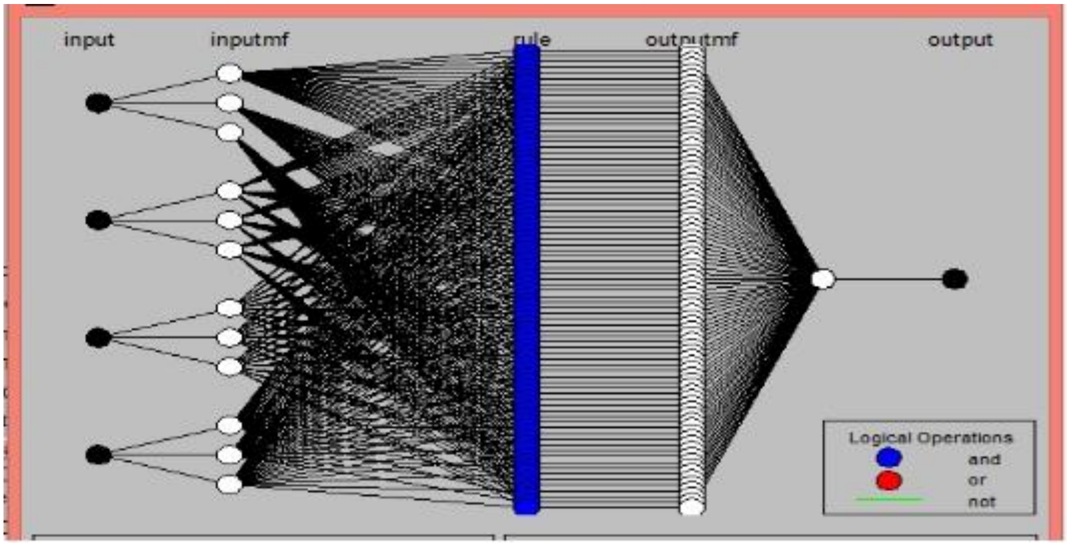


Figure 6. 3: Neuro-fuzzy system structural (4 inputs features, 3 membership function, 81 rules and one output).

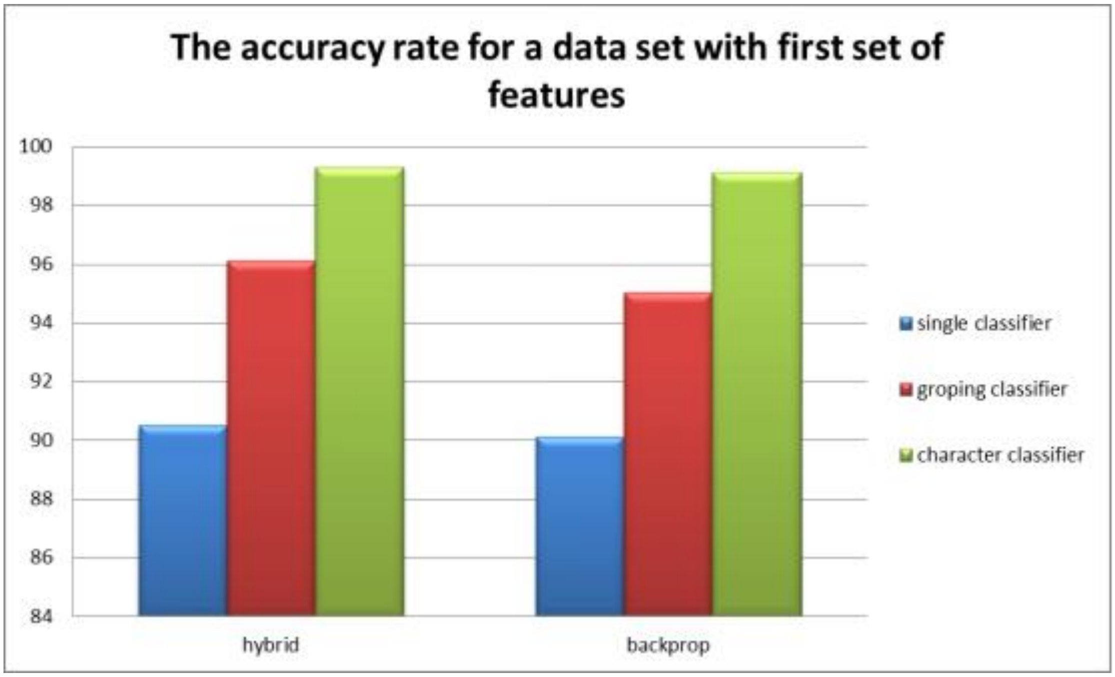


Figure 6. 4: Accuracy rate for a testing data set with CCOB of features.

6.4 Experiments with Second Statistical Set of Features using PCA Technique

The handwritten character images are taken from the SUST-ARGG data set. The outer parts of these images are removed using the cropping technique. After the unwanted background parts of the image are omitted, the cropped images are resized to 7 by 5. The images are then passed to the feature extraction process in order to calculate the score values using the principal component analysis (PCA) technique. The score values which were obtained from the PCA techniques are then used by ANFIS classifier for doing the training process. Figure 6.5 shows the second set of feature classifier.

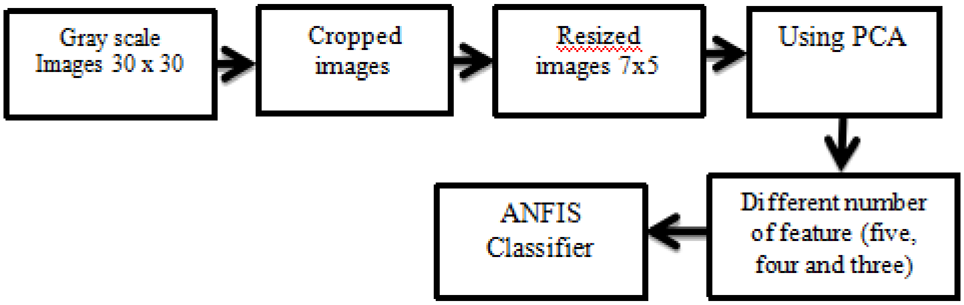

Gray scale Images 30 x 30
Cropped images
Resized images 7x5
Using PCA
Different number of feature (five, four and three)
ANFIS Classifier

Figure 6. 5: Second set of features diagram.

Five, four and three features were used as inputs to the system and three numbers of triangular membership function was also used for each and single output. Figure 6.6 show the ANFIS structure, with Five features were used as inputs to the system, three numbers of triangular membership function were used, one output .

A set of different experiments were applied on the training data set and the testing data set for all types of data set with different numbers of features. The results are reported in tables 6.3, 6.4 and 6.5, respectively.

Table 6. 3: Accuracy rate for the data set with 3 features.

Dataset type	Training accuracy %		Testing accuracy%	
	Hybrid	Backprop	Hybrid	Backprop
All data set (single classifier)	91.03	90.38	**90.96**	90.36
Grouping data set (grouping classifier)	96.13	95.81	**95.96**	95.70
Similar data set (characters classifiers)	99.52	99.47	**99.46**	99.42

Figure 6.7 shows the accuracy rate for all levels of the data set using 3 features extracted by the PCA technique. The figure also shows that the accuracy for the two learning algorithm gets marginally better recognition accuracy. The hybrid rate results are 95.96% and 99.46% compared to the back-propagation rate of 95.70% and 99.42% for grouping classifier and characters classifier, respectively.

Figure 6.8 shows the accuracy rate for all levels of data set using 4 features extracted by the PCA technique. The figure also shows that the accuracy of the two learning algorithm gets better recognition accuracy. The hybrid rate are 96.2% compared to the back-propagation rate of 95.1% for grouping and characters classifier, respectively.

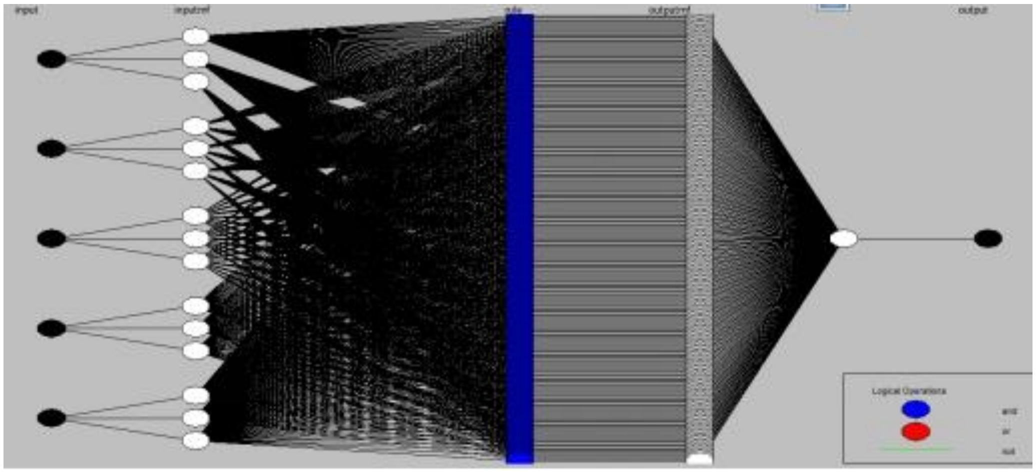
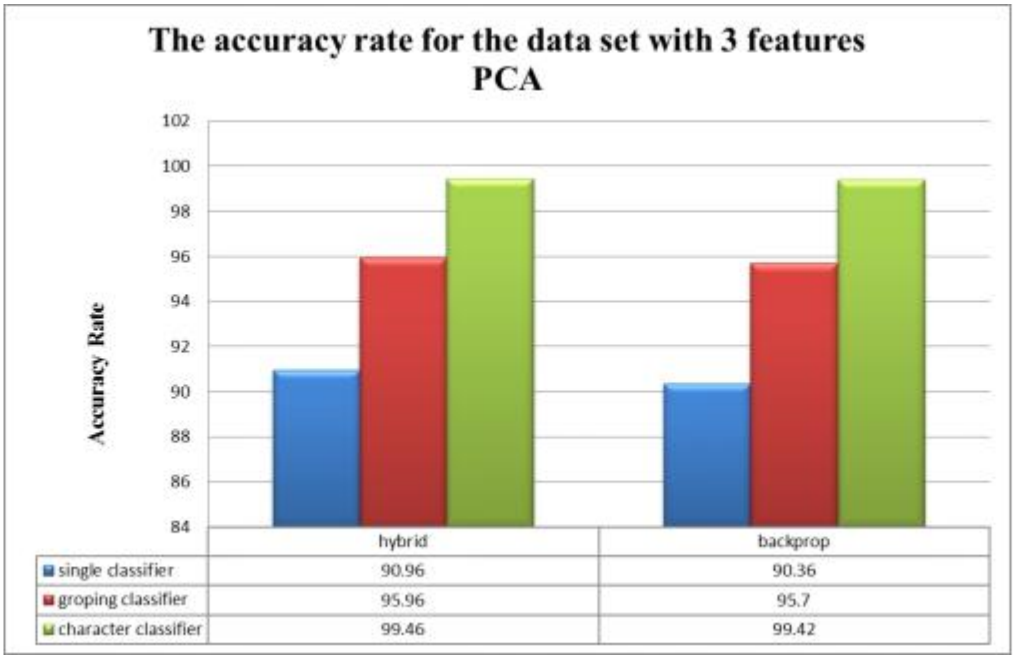


Figure 6. 6: Neuro-fuzzy system structural (5 inputs as features, 3 membership function, 405 rules and one output).



	hybrid	backprop
🟦 single classifier	90.96	90.36
🟥 groping classifier	95.96	95.7
🟩 character classifier	99.46	99.42

Figure 6. 7: Accuracy rate for the testing data set with 3 features PCA.

Table 6. 4: Accuracy rate for the data set with 4 number of features.

Dataset type	**Training accuracy %**		**Testing accuracy%**	
	Hybrid	Backprop	Hybrid	Backprop
All data set (single classifier)	91.59	90.40	**91.39**	90.38
Grouping data set (grouping classifier)	96.4	95.6	**96.2**	95.1
Similar data set (characters classifiers)	99.62	99.2	**99.38**	99.12

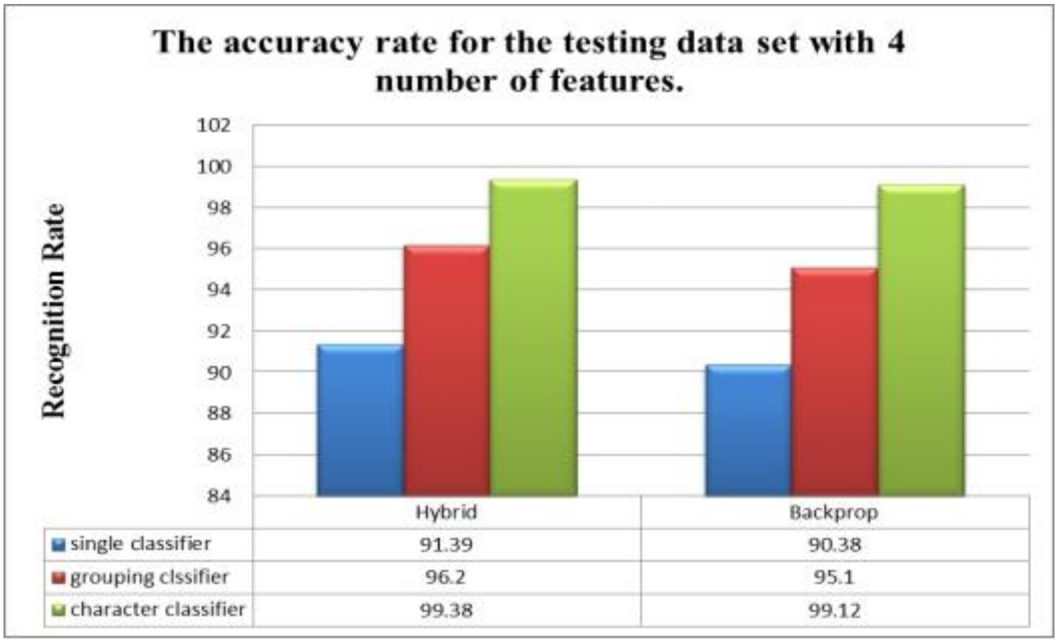


Figure 6. 8: Accuracy rate for the testing data set with 4 features PCA.

Table 6. 5: Accuracy rate for the data set with 5 number of features.

Data set type	**Training accuracy %**		**Testing accuracy%**	
	Hybrid	Backprop	Hybrid	Backprop
All data set (single classifier)	92.58	hunt	**92.11**	hunt
Grouping data set (grouping classifier)	96.75	hunt	**95.94**	hunt
Similar data set (characters classifiers)	99.70	hunt	**98.26**	hunt

The recognition rate of character recognition calculated by using hybrid method and using the back-propagation learning algorithm are shown in table 6.2, table 6.3, table 6.4 and table 6.5. The capability of the ANFIS model in recognition rate is better using the hybrid learning algorithm than when using the back-propagation method. This is because the hybrid method comprises of back propagation and Least-Square methods. The proposed system has reached its limit after two stages.

With the second set of features, PCA technique was used to reduce the dimensionality of the feature vectors to 3, 4 and 5 for any 35 size of the character cropped images.

In single classifier for 34 characters data set as shown in table 6.3, table 6.4 and table 6.5, the highest recognition rates were 90.96%, 91.39% and 92.11% for 3, 4 and 5 features, respectively. We noted that there is an improvement in the recognition result whenever we used more features, but the training process needed more memory and time.

In group classifier results for 15 groups of character data set as shown in table 6.3, table 6.4 and table 6.5, the highest recognition rates were 95.96%, 96.20% and 95.94%

with 3, 4 and 5 features, respectively. The highest recognition rate was observed when the middle number of features was used.

In character classifier results for a similar group of character data set as shown in table 6.6, the highest recognition rates were 99.46%, 99.38% and 98.26% with 3, 4 and 5 features, respectively. The highest recognition rate was observed when 3 features was used. Whenever a large number of features were used, then the average rate of recognition for testing data set was low as shown in table 6.6.

In the second stage classifiers, as shown in table 6.6 there are differences in the recognition rates for the classifier with different number of features. For example, the recognition rate for Group 4 with 4 features has the best result.

Table 6. 6: Accuracy for the characters classifier with different features using PCA.

Similar data set Groups	3 Features	4 features	5 features
Groups 1	99.46	99.43	98.93
Group2	99.26	99.33	98.06
Group3	99.31	99.38	97.23
Group4	99.30	99.40	99.02
Group5	99.37	99.22	99.00
Group7	99.48	99.23	98.59
Group8	99.71	99.21	99.23
Group9	99.40	99.30	98.99
Group10	99.65	99.72	99.37
Group14	99.65	99.60	99.23
average	**99.46**	**99.38**	**98.26**

Figure 6.9 shows a comparison between the results of the three number of features extracted with PCA. The results show that the 3 number of features gave a better recognition rate than the other number of features.

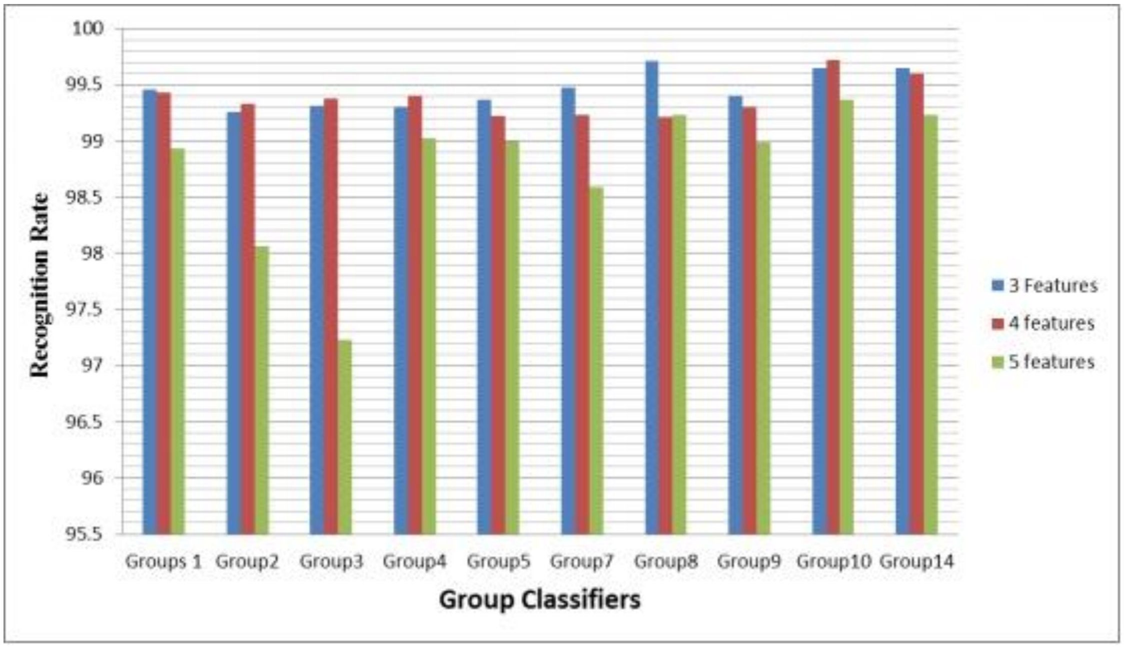


Figure 6. 9: Accuracy for the characters classifier with different number of features, 3, 4, and 5 using PCA.

6.5 Accuracy Rates for Classes with Four Features.

Table 6.7 shows the recognition rate for grouped class in the first stage. Table 6.8 shows the recognition accuracy results for any class that was obtained with single ANFIS classifiers for all data set. Table 6.9 shows the percentage recognition for character classes in the second stage.

Theorem1: Unique Features

The maximum accuracy for Handwritten Arabic character recognition with unique features using the proposed approach, two stages ANFIS classifier, is 100%.

Proof: Based on the simulation results in table 6.8, recognition rate 100% for the recognized characters such as (ط), because it consist of loop, vertical, horizontal line and has no dots. Because all of these features are not found in another character.

Theorem2: High Similarity Features

The accuracy for handwritten Arabic character recognition with high similarity features using this proposed method (classes with four features) is low.

Proof: The proof is based on the simulation. The simulation results in table 6.8 show the following:

1. The recognition rate for letters like (ي, ئ) is 82.40% and 88.8% respectively. The accuracy rate for letter (ئ) is low because of the extreme similarity to the letter (ي).

2. The recognition rate for letters like (ب, ت and ن) is 88.83%, 89.40 and 87.16% respectively. The accuracy rate for letter (ب) is low because of the high similarity of the letter (ب) with letters (ت and ن).

The simulation result in table 6.7 shows that the accuracy rate for the group (ي and ئ) is low among other groups of grouping classifier because of the high similarity of the (ي and ئ) with letters of group (ب, ت and ن).

Table 6. 7: Accuracy for the 15 classes of characters (groups) with testing dataset.

Accuracy	Group	N	Accuracy	Group	N
97.66	ع غ	8	96.13	ا إ أ	1
97.9	ف ق ؤ	9	95.36	ب ت ث ن	2
97.28	ل ك	10	96.06	ج ح خ	3
97.45	م	11	95.92	د ذ ر ز ء	4
97.99	هـ	12	96.33	س ص ض	5
96.68	و	13	98.54	ش	6
93.90	ئ ي	14	99.21	ط ظ	7
93.96	لا	15			

Table 6. 8: Accuracy for the 34 classes of characters.

Accuracy	characters	N	Accuracy	characters	N
94.36	ع	18	91.94	ا	1
94.65	غ	19	87.16	ب	2
95.39	ف	20	89.40	ت	3
96.97	ق	21	90.14	ث	4
96	ك	22	90.41	ج	5
95.13	ل	23	92.16	ح	6
94.12	م	24	92.11	خ	7
88.83	ن	25	88.46	د	8
95.89	هـ	26	93.6	ذ	9
94.90	و	27	91.13	ر	10
88.08	ي	28	96.66	ز	11
91.78	لا	29	95.87	س	12
92.76	ء	30	96.61	ش	13
91.61	أ	31	95.98	ص	14
94.73	إ	32	97.09	ض	15
85.21	ؤ	33	100	ط	16
82.40	ئ	34	98.23	ظ	17

Table 6. 9: A sample accuracy rate for group two (N=2) that contains similar characters.

Accuracy recognition	Characters	N
99.86	ب	1
98.96	ت	2
98.53	ث	3
98.88	ن	4

Table 6.9 is a sample of the accuracy rate well using the second stage classifier. The result shows there is improvement in the recognition rate when the second stage is used. For example, the recognition rate for letter “ت” is 89.40% when single classifier is used as shown in table 6.8 and it is 98.96% when the second stage classifier is used.

Figure 6.10 shows a comparison between the results of the 34 features extracted with PCA. The result shows that the Ta “ط” letter gets a best recognition rate of 100%. Also shows the “ئ” letter get a low recognition rate of 82.40%.

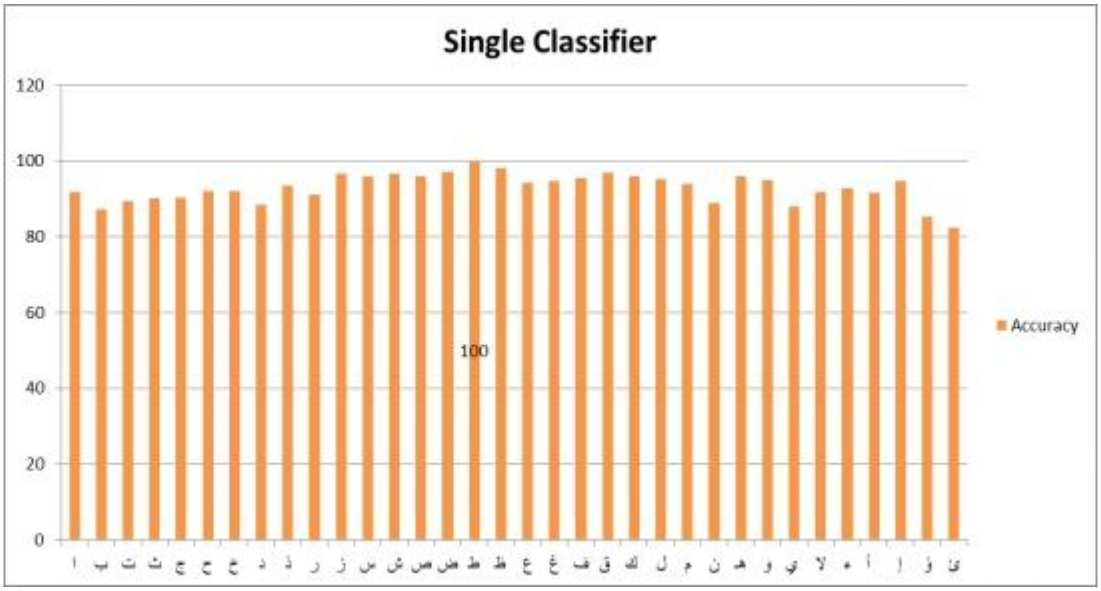


Figure 6. 10: Accuracy for the 34 classes of characters with single classifier.

6.6 Experiments with Hybrid Features (CCOB and PCA)

The handwritten character images are taken from the data set. These images are first, Binarized and then morphology is used on them. The second pass to the feature techniques extract four features as CCOB features. With the second set of features, the image characters are cleaned by removing its outer parts using the cropping technique. After the unwanted background parts of the image are omitted, the focused was then on the number and resize of the cropped images to 7 by 5. The two set of features (4+35) were combined and reduced to four features used the PCA technique. Then four feature values which were obtained from the PCA technique were used by the ANFIS classifier for performing the training process. Figure 6.11 shows the combined features process.

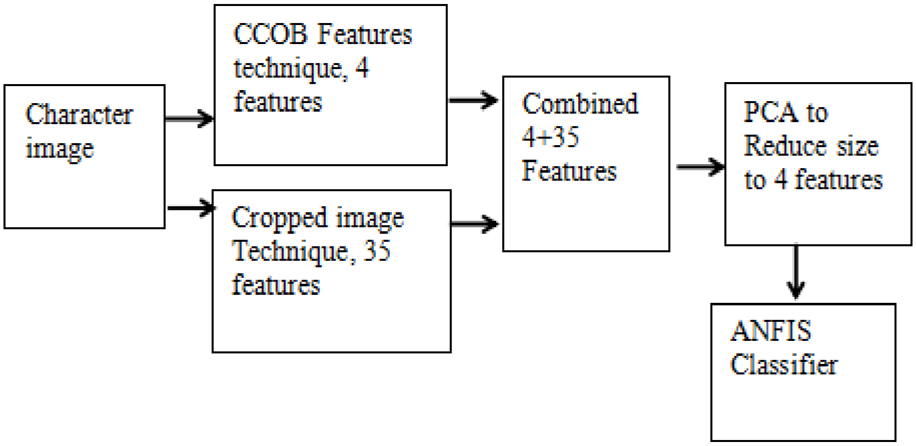

Character image
CCOB Features technique, 4 features
Cropped image Technique, 35 features
Combined 4+35 Features
PCA to Reduce size to 4 features
ANFIS Classifier

Figure 6. 11: The proposed classification system steps with combined features.

Four features were used as inputs to the system, three number of triangular membership function were also used, and single output with 1000 epoch was performed. A set of different experiments were performed on the training data set and the testing data set for all type of data set as shown in table 6.10.

Table 6.11 shows the recognition accuracy results for any character classifier, which were classified in the second stage.

The proposed system has improvement the first stage with a recognition accuracy of 97.15% for the isolated Arabic handwritten character in grouped classifiers and 99.34% recognition rate with the second stage with an increase of about 3.5% from the recognition accuracy achieved by a single classifier system as shown in table 6.10.

Table 6.10: Accuracy rate for the all data set with hybrid set of features.

Dataset type	Training accuracy	Testing accuracy
All dataset (Single classifier)	95.82	95.82
Grouping dataset (Grouping classifier)	97.18	97.15
Similar dataset (Character classifier)	99.6	99.34

Figure 6.12 shows the accuracy rate of the all levels of data set using hybrid features extracted by (PCA and CCOB) technique. The figure also shows the high accuracy with single and grouping classifiers 95.82% and 97.15, respectively compared to the previous experiments with PCA and CCOB Separately.

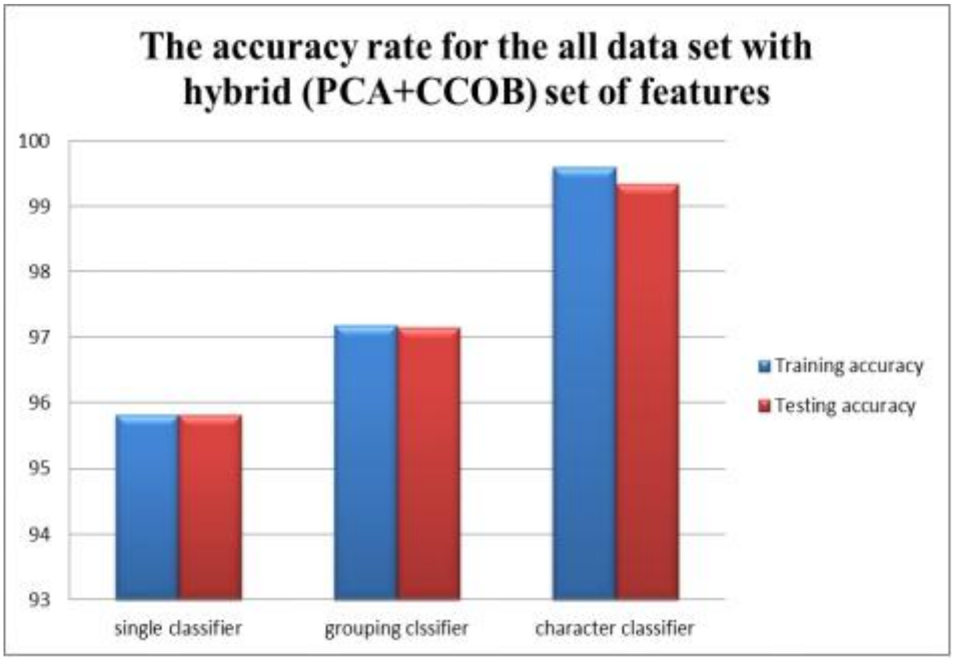


Figure 6. 12: Accuracy rate for the training and testing data set with hybrid features extraction technique.

Table 6.11: Accuracy for the characters classifier with combined features.

Classifier groups	Training accuracy %	Testing accuracy%
Groups 1	99.66	99.42
Group2	99.34	99.21
Group3	99.52	99.23
Group4	99.21	99.40
Group5	99.59	99.32
Group7	99.63	99.13
Group8	99.78	99.05
Group9	99.51	99.35
Group10	99.98	99.70
Group14	99.77	99.60

6.7 Addition Experiments with Different Grouped Data Set

10 grouped similar character (character-10) classes were selected from SUST-ARG data set as shown in table 6.14 and conducted experiments with a two set of features. The result are shown in table 6.12 which shows that there is improvement in the recognition rate when compared to the grouped similar character-15 classes in previous experiments as shown in table 6.2 and table 6.3, the result for grouping data set.

Table 6. 12: Accuracy rate for the grouped character-10 classes data set.

Features type	Training accuracy %		Testing accuracy%	
	Hybrid	Backprop	Hybrid	Backprop
CCOB	97.58	97.42	96.83	95.22
PCA	98.04	97.75	**97.49**	97.48

Table 6.13 shows the accuracy rate for group class 2 which contains the letters of (ي ئ ب ت ث ن), among the grouped characters data set (charcter-10 classes) at the second stage of classification the recognition rate for testing the dataset is 98.88%. For the group

class in the previous character-15 classes' data set which contain the letters (ب ت ث ن), the recognition rate for testing data set obtained is 99.2%. The result indicates that there is little improvement in terms of accuracy for the separate group of characters.

Table 6. 7: Accuracy rate for the group 2 with character-10 classes.

Features type	Training accuracy %		Testing accuracy%	
	Hybrid	Backprop	Hybrid	Backprop
PCA	99.00	98.78	98.88	97.60

Table 6. 8: Grouped similar character-10 classes and character-15 classes.

Group number	Ten Character Group	Fifteen Character Group
1	أ إ ا م	أ إ ا
2	ي ئ ب ت ث ن	ب ت ث ن
3	ع غ ج ح خ	ج ح خ
4	د ذ ر ز ء	د ذ ر ز ء
5	س ش ص ض ق ف	س ص ض
6	لا ط ظ	ش
7	و	ط ظ
8	ؤ	ع غ
9	ل ك	ف ق ؤ
10	هـ	ل ك
11		م
12		هـ
13		و
14		ئ ي
15		لا

Theorem 3: Grouping

The best accuracy achieved for the isolated Arabic handwritten character recognition (IAHCR) is for X group classifier, where 1<= X < =34.

Proof: The proof is based on the simulation. The simulation results in table 6.2 and 6.3 shows that the recognition rate for testing data set with different grouping levels. For example, when the IAHCR contained 15 classes the accuracy was 95.81% and 95.02% with PCA and CCOB features extraction, respectively. The recognition rates for testing data set with grouping level which contains 10 classes was 96.83% and 97.49% with self-features extraction as shown in table 6.12.

The results show that the accuracy rate for grouping-10 classes is better than grouping-15 classes. The result shows that whenever a data set is divided into less groups then the rate of recognition is better.

6.8 Time and Space Analysis

The feature extraction techniques and classifier methods presented in chapter two and studied in chapter three and proposed in chapter four respectively, and implemented with new two stage proposed of AHCR using ANFIS classifier in this chapter.

The performance of the ANFIS classifier is evaluated in this chapter, the previous sections described the results of experiments with recognition rate for any level with different extracted features. In addition, to evaluate the efficiency of the proposed methods would involve analyzing other aspects such as processing time and storage space, which are explained in Section 6.8.1 and 6.8.2, respectively.

6.8.1 Processing Time

The calculation of the recognition time is important for the efficiency of the proposed methods. Time evaluation of the recognition system depends on the experiments which were conducted on the data sets for the AHC. This involves calculating the processing time for different feature extraction methods that used and training time for the data sets.

6.8.1.1 Time processing for Feature Extraction Methods

For comparison with the applied feature extraction methods, three types of feature extraction (FE) methods such as Morphology with CCOB features, cropped with PCA features and Hybrid features as presented in sections 6.3, 6.4 and 6.5 are selected, respectively. The simulations of recognition results of features extraction phase are to evaluate the efficiency of classification. As shown in table 6.2, table 6.3, table 6.4, table 6.5 , and table 6.10.

Table 6.15 explains the processing times for these feature extraction methods. It is the time consuming to compute the extracted features from different sets of database.

The average simulation time consumed by the extraction of a single handwritten character to FE the PCA, CCOB, and Hybrid is 0.011, 0.017, and 0.022 seconds, respectively, as shown in table 6.15, which calculated by dividing the time of features extraction for the data set on the number samples in the data set. The hybrid FE method took the longest time due to the needed processing steps for extracting the features separately, combined and then reduce them to four features.

Figure 6.13 shows the comparison of timing analysis for three types of features extraction technique. The figure shows that the PCA FE method took the shortest time for all levels of classifiers.

Table 6. 9: Result of timing analysis for the different features extraction methods.

Data set type	**FE time (second)**				
	PCA			CCOB	Hybrid (PCA+CCOB)
	3 features	4 features	5 features		
All data set (Single classifier)	71.8	72.4	73.18	117.04	147.8
Grouping data set with 15 classes (Group classifier)	32.3	31.8	30.6	52.37	89.18
Similar data set (Character classifier)	4.43	4.2	4.51	10.4	17
Average time for single handwritten character	**0.011**			**0.017**	**0.022**

Table 6.15 shows if the number of classes in the data set is large, then the time for feature extraction is also large. For example, if the average time for features extracted with PCA technique is calculated then the time spent to extract features for all the data set which contains 34 classes is 72.4 seconds, for the grouping data set with 15 classes is 31.8 seconds and finally 3 seconds for the character class. Also, the table shows that if the number of extracted features is large then the time is high. For example, in experiments using PCA, the time when extracted for 3 features took 32.3 seconds and for 4 features was 31.8 seconds.

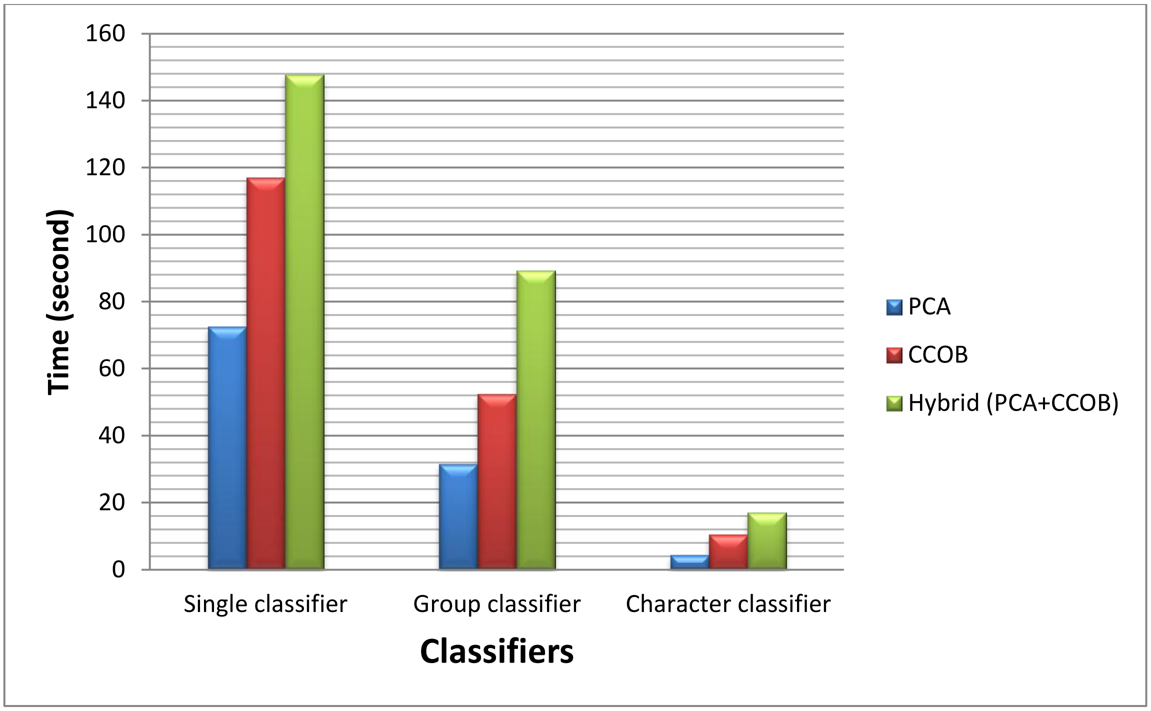


Figure 6. 13: comparison of timing analysis for the different features extraction methods.

The time that it took to extract features with the CCOB was more than the time for PCA because the simulation is more complex with CCOB than in the PCA. In the hybrid features extraction the time is almost equal to the time for the two types combined.

The average simulation time consumed by the extraction of a single handwritten character as shown in table 6.15 show that the hybrid FE method took the longest time due to the needed processing steps for extracting the features separately, combine them, and then reduce them.

6.8.1.2 Time Processing for Training Data set

Neural network training time is the time that has been spent in the data set training. It is the time calculated from the start of training process with first epoch until the system reaches the proper recognition rate. The training time is a parameter independent of training data set and is the number of minutes spent training a particular network.

Table 6.16 and table 6.17 show the training time in minutes for ANFIS classifier with different features. These tables show that the training time spent with hybrid learning algorithm is more than the training time spent with back-propagation learning algorithm. This is due to the hybrid algorithm optimizing two types of parameters, the premise and consequent for backward and forward, respectively with ANFIS operations were the back-propagation adjusts only premise parameters.

Tables 6.16 and 6.17 also show the average simulation time consumed by the training process of a single handwritten character.

Tables 6.16 and 6.17 show that whenever the number of features extracted by PCA technique used with the ANFIS training process is large then the time spent is more.

Table 6. 10: Result of timing analysis for the training process in ANFIS classifier with hybrid learning algorithm and different feature extraction methods.

Data set type	Training time (minutes)				
	PCA			CCOB features	Hybrid (PCA+CCOB)
	Three features	Four features	Five features		
All data sets (Single classifier)	9.55	35.11	383.20	35.17	35.4
Grouping data set (Group classifier)	3.40	15.14	133.51	21.17	15.16
Similar data set (Character classifier)	0.52	3.33	28.19	0.56	3.59
Average time for training single handwritten character	0.0019	0.0068	0.075	0.0069	0.0067

Table 6.17: Result of timing analysis for the training process in ANFIS classifier with back-propagation learning algorithm and different feature extraction methods.

Data set type	Training time (minutes)			
	PCA		CCOB features	Hybrid (PCA+CCOB)
	three features	four features		
All data set (single classifier)	8.23	25.26	25.58	24.46
Grouping data set (group classifier)	3.15	11.14	14.37	10.40
Similar data set (character classifier)	0.50	2.44	0.45	1.48
Average time for training single handwritten character	0.0016	0.005	0.0057	0.0047

6.8.1.3 Recognition Time

Total time to recognize the character with one stage with PCA, CCOB and Hybrid features extraction is 11.67, 17.67 and 22.67 millisecond respectively, for each character as shown in table 6.18. The total time to recognize the character with two stages using PCA, CCOB and Hybrid features is 12.34, 18.34 and 23.34 millisecond respectively, as shown in tables 6.18. The result shows that the recognition with PCA technique uses less time and recognition with Hybrid feature technique spent more time.

Table 6.18: Result of timing for recognition for each character in the proposed recognition system.

Recognition phase	**Average processing time for per character (milliseconds)**		
	PCA	CCOB	Hybrid
Features extraction	11.00	17.00	22.00
Recognition	0.67	0.67	0.67
Full time (one stage)	11.67	17.67	22.67
Full time (two stages)	12.34	18.34	23.34

6.8.2 Storage Space

System recognition to be a desirable in the market should be characterized by small storage. So that basic indicator for determine the required storage space for the recognition system are the size of the extracted features. This is described below.

Table 6.19 shows the comparison between the different extracted features, which include CCOB, PCA and HYBRID. The space storage is computed using the follows [100]:

$$\text{Feature size} = (dn)^2 \qquad (6.1)$$

Where d is the features number and n is the samples number. Table 6.19 shows the features size as following:

1. The space for storing 5100 sample is 9.302 MB for three features and the space is 15.51 MB for five features.

2. For storing 2250 samples, the space is 5.431 MB for three features and the space is 15.087 MB for five features.

3. For storing similar characters, the average space is 0.485 MB for three features and the space is 1.896 MB for five features.

4. The average simulation storage space for a single handwritten character is 19.92 bytes.

Table 6.19: Feature sizes for the three different feature extraction methods.

Data set type	**Features size (MB)**				
	PCA			CCOB features 4 features	Hybrid (PCA+CCOB) 4 features
	Three features	Four features	Five features		
All data sets 5100	9.302	12.4	15.51	12.4	12.4
Grouping data set 2250	5.431	9.656	15.087	9.656	9.656
Similar data set (300 to 750)	0.485	0.948	1.896	0.948	0.948

6.9 Comparative Study

A comparative study for the proposed idea has been presented and compared with the method presented by balola, shaout and elhafiz [101]. In this comparative study, we compared the neuro-fuzzy proposed system with the neural network system designed for Arabic handwritten character recognition. Table 6.20 shows that all the result of simulation have been evaluated on the same size of SUST-ARG data sets. The comparison shows the effectiveness of our proposed system. In [89] authors have proposed an offline character recognition system for isolated Arabic alphabet written by a single writer. They obtained 97% recognition accuracy with five stages. If compared this result with the proposed system for grouped data set of SUST-ARG data set which was written by 141 person, then the recognition accuracy shows an improvement using our proposed system as shown in table 6.20.

Table 6.20: Comparison results of proposed method with other methods

Authors	**Technique**	**Testing accuracy%**		
		All dataset	Grouping dataset	Similar dataset
Proposed method	Neuro-fuzzy	95.8173	**97.49**	**99.46**
Balola, Shaout and Elhafiz [100]	Neural Network	54.3	**78.77**	**92.77**
Randa E, M. A. Rashwan abd Samia A. M.[101]	Decision Fusion	74.48	96	98.6

6.10 Discussion

The new proposed method for Arabic recognition presented in this dissertation has divided the problem into multiple sub-problems. The method is made of a two stage classification system. The two-stage classifier separates the classification problem into different modules. To improve the performance of recognition result, the characters are divided into multi-groups using some knowledge about similarities among the characters. The design of the classifier is made based on both the similarities among character structures and a multistage classification. The design of the handwritten Arabic character classifiers is a multistage classifier that has two stages. The first stage is based on features extracted from each group of similar characters that were placed in separate classes. The last stage has sub-classifiers each sub-classifier is a multiple parallel classifier. The final decision is made by calculating the largest value of the averaged outputs from the two network stages.

In the approach, two types of statistical features have been employed. The first type of features, the area that is formed from the projections of the upper and lower, as well as of the left and right character profiles is calculated, the center mass (xt, yt) of the character image , count the number of transitions, Outliers (Right, Left, Top and Down) and Black ink histograms. The second set of features is based on the extract eigenvectors and eigenvalues of shifting meaning of resize cropped images.

The proposed method used cropped grayscale images of Arabic handwritten character, with PCA features technique and morphology binarized images with the second set of features. Neuro-fuzzy classifier used as a classification method with hybrid and backpropagation learning algorithms. In the experiments conducted, the data set was divided into three levels. Within a level, the characters are divided into multi-groups using some knowledge about the similarity among the characters. The proposed classification system is made of a two stage classification to improve the performance of recognition rate.

Table 6.21: Highest accuracy rate for the all levels of classifiers with different types of features.

Classifiers levels	**Testing accuracy%**		
	CCOB	PCA	Hybrid (CCOB+PCA)
Single classifier	90.5	92.11	**95.82**
Group classifier	96.1	96.2	**97.15**
Character classifier	99.30	**99.46**	99.34

Table 6.21 shows the highest recognition rate for the experiments tested for all levels of classifiers using different set of features. Results show that the best performance for single classifier is 95.82%, group classifier is 97.15% with Hybrid of features and the average of characters classifier is 99.46% with PCA obtained for testing data set. Figure 6.14 shows the comparison of the three types of features. All these experiments used with the hybrid learning algorithm.

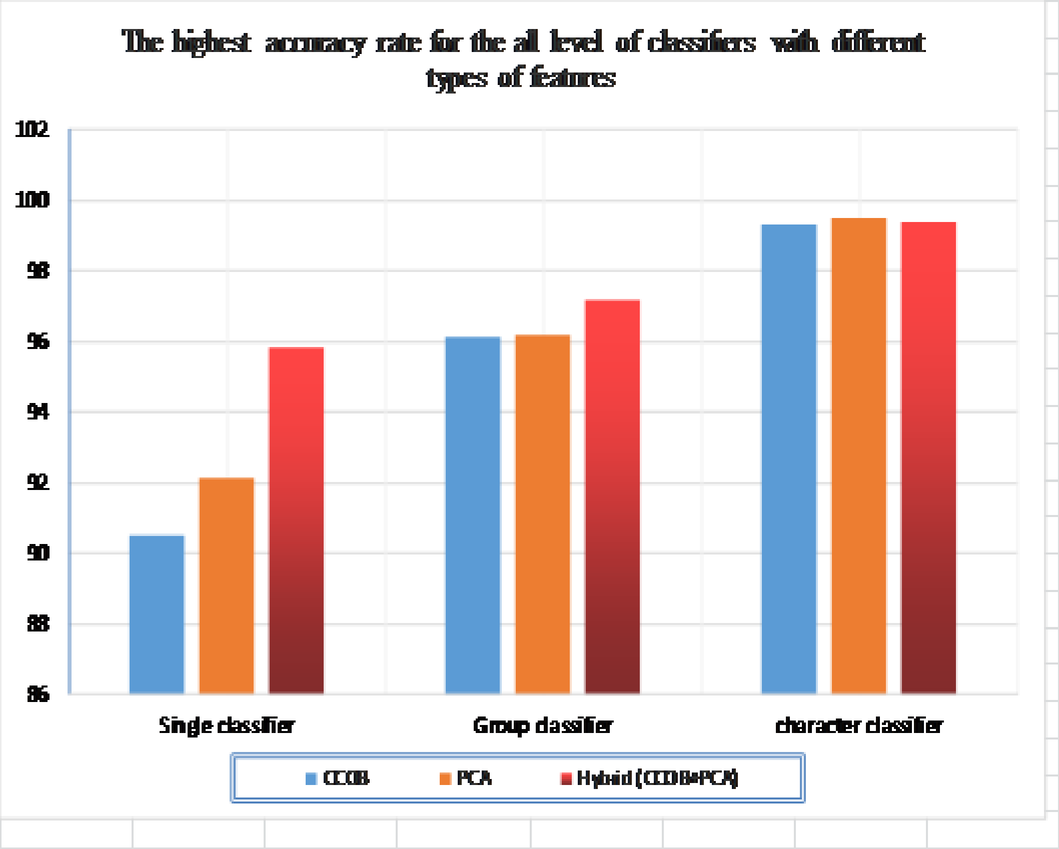


Figure 6. 14: Comparison with different features and levels.

Based on the results shown in table 6.2, table 6.3 and table 6.10 the small differences between the training and testing result suggest that a good generalization has been achieved.

Experimental results have shown that the ANFIS with hybrid learning algorithm is an effective technique for improving character recognition accuracy rate.

After examining the recognition accuracy of each character we found that the recognition rate is 100% for recognizing characters such as (ط). The recognition rate for letters like (ئ ,ي, ب, ت, ن) is low as shown in table 6.8, because they all look very similar in shape. In table 6.7, grouped classifier found that the recognition rate was 93.90% for character group such as (ي ,ئ) which is low. This shows that the lowest rate in the first stage refers to the presence of the dot and "Hamza "in other letters.

Recognition rates of different feature extraction methods are also analyzed in this dissertation.

In all previous experiments with different features and different size of features there are improvement when second stage is used, compared to only using a single classifier. In addition, for the second stage, a group can use separate feature-classifier than those used with another group. Thus, the proposed method is more flexible and efficient than single classifier.

6.9 Summary

In this chapter Neuro-Fuzzy classifier with different learning algorithm for two stages of classification of Arabic handwriting character recognition for both grouped and separate characters has been proposed. Two set of features and combined features has been used. Some experiments were presented for character recognition system.

SUST-ARG data set has been used and a set of experiments was conducted on the system. The results of these experiments have shown a good performance.

CHAPTER VII

CONCLUSION AND FUTURE WORK

In this dissertation, a system for recognizing handwritten isolated Arabic characters using a two-stage neuro-fuzzy system with different learning algorithms based scheme using two set of feature techniques has been presented. The data set that is used in these experiments is from isolated Arabic handwritten character set. It was collected from 141 writers with each person writing each letter ten times and scanned with 300 dpi.

Two types of statistical features have been employed. The first type of features (CCOB) includes the Center mass (xt, yt) of the character image, (Cross hair) count the number of transitions, Outliers (Right, Left, Top and Down) and Black ink histograms. The second set of features is based on the extract mean of resize cropped images, calculated covariance, computed eigenvectors and eigenvalues of shifted. We used hybrid features of existing types. The number of features, which were extracted, are as follows: 4 features for the first set, 3, 4 and 5 features for the second set and 4 features for the hybrid model.

Many experiments have been conducted in this research. The experiments used 34, 15 and (1 to 5) character classes. The highest result accuracy for testing data set with first stage (15 classes) was **97.15%** as shown in table 6.10, with hybrid features. For the second stage (character classifier) we had a rate of recognition of **99.46**% as shown in table 6.3 with 3 features that were extracted by PCA technique. The recognition rate results show that there was an improvement in the performance of the proposed system.

The results in tables 6.2 and 6.4 have shown that the feature extraction based on PCA technique has better recognition rate than the first statistical features based on CCOB technique.

When collecting large number of letters that have similar structure then there was an improvement in the recognition rate with the first stage, but the second stage may not be the best. Improved results were achieved when the proposed two stages system was used compared to what other researchers in the character recognition field used.

In the further research work, we would like to improve the recognition accuracy of neuro-fuzzy for handwritten Arabic character recognition by doing the following: using more training samples, using different data set to further validate the system and using a good feature extraction techniques and different membership functions.

References

[1] F. Hussain and J. Cowell, “Character Recognition of Arabic and Latin Scripts”, IEEE International Conference on Information Visualization, 2000, pp. 51–56.

[2] A.A. Zaidan, , B.B. Zaidan, H.A.Jalab, H.O.Alanazi and R. Alnaqeib, “Offline Arabic Handwriting Recognition Using Artificial Neural Network”, journal of computer science and engineering, vol. 1, Issue 1,pp. 47-67, May, 2010.

[3] A. L. Koerich, R. Sabourin and C. Y. Suen, “Large vocabulary off-line handwriting recognition: A survey”, journal of Pattern Anal Applic, pp. 97–121, jun. 2003.

[4] H. A. Abed, “Fuzzy Logic approach to Recognition of Isolated Arabic Characters,” International Journal of Computer Theory and Engineering, Vol. 2, No. 1, pp.9-17, February. 2010.

[5] R. E, Introduction to Fuzzy set theory basic concepts and structures, IEEE, USA, 1992.

[6] Z. A. Zadeh. Fuzzy logic systems: Origin, concepts, and trends. Computer Science Division Department of EECS UC Berkeley. 2004 Nov 10.

[7] J. A. Freeman and D. M. Skapura. Neural Networks Algorithms, Applications, and Programming Techniques. Addison-Wesley Publishing Company, Redwood City, CA, USA, 1991.

[8] S. Theodoridis. Pattern recognition second edition. Elsevier, (USA), 2003.

[9] K. Fukunaga. Introduction to statistical pattern recognition (2nd ed). Academic Press, Boston 1990.

[10] T. Pavlidis. Structural Pattern Recognition. Springer Verlag, Berlin Heidelberg New York, 1977.

[11] R. P.W. Duin,”Structural, Syntactic, and Statistical and Pattern Recognition “, Joint Iapr International Workshops Windsor, Ontario, Canada, August 6-9, 2002.

[12] V. Dutt, V. Chaudhry, I. Khan. "Pattern recognition: An overview", American Journal of Intelligent Systems, vol. 2, pp 23-27, 2012.

[13]. C. Y. Suen,” Character recognition by computer and applications”, In Pattern Recognition and Image Processing, T. Y. Young and K. S. Fu, editors, Academic Press, Orlando, FL, 1986, pp. 569–586.

[14] R.Bellman, R.Kalaba and L.A.Zadeh, “Abstraction and pattern classification”, Journal of Mathematical Analysis and Applications, vol. 13, pp. 1-7, 1966.

[15] A. k. sharma and B. V. Padamwar,”Fuzzy Logic Based Systems in Management and Business Applicaions”, International Journal of Innovative Research in Engineering & Science, vol. 1 issue 2, January. 2013.

[16] R. Balamondon and S. N. Srihari, “On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, Jan. 2000.

[17] A. Muhammad, H. Hassan, D.Nasien and H. Haron, "A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 6, No. 2, 2015.

[18] N. Arica, and F.T. Yarman-Vural, “An Overview of Character Recognition Focused on Off-line Handwriting”, IEEE Trans. On Systems, Man, and Cybernetics, Vol. 31. No. 2, pp. 216-233, 2001.

[19] A. Ashraf, “Offline printed Arabic character recognition”, PhD thesis, University of Nottingham, 2012.

[20] R. Plamondon and S. N. Srihari, “On-line and off-line handwritten recognition: a comprehensive survey”, IEEE Transactions on PAMI, Vol. 22(1), pp. 63–84, 2000.

[21] A. Amin, “Arabic Character Recognition,” in Character Recognition and Document Image Analysis, World Scientific Publishing Company, 1997, pp. 398.

[22] M. Cheriet, “Strategies for Visual Arabic Handwriting Recognition. Issues and Case Study”, IEEE Int. Conference on Signal Processing and Applications, Sharjah (UAE), 2007.

[23] K. Daqrouq, M. N. Ajour, A. Alkhateeb, A. Morfeq, A. Dobaie, M. Badarin, and A. Rihawe, " Handwritten Arabic Characters Recognition Based on Wavelet Entropy and Neural Network", British Journal of Applied Science & Technology, vol. 9(5), pp. 464-474 , 2015.

[24] Sh. Abdelazeem, H. M. Eraqi, H. Ahmed, “On-line Arabic Handwritten Word Recognition Based on HMM and Combination of On-line and Off-line Features”, in Guide to OCR for Arabic Scripts, Springer London, 2012, pp 559-585.

[25] S. Impedovo,"Fundamentals in Handwriting Recognition", Springer- Verlag Publ., Berlin, Volume 124, 1994.

[26] I.S. Abuhaiba, S.A. Mahmoud and R.G. Green,” Recognition of Handwritten Cursive Arabic Characters”. IEEE Trans. PAMI 16(6), pp. 664–672, 1994.

[27] T. Buckwalter, “Issues in Arabic Orthography and Morphology Analysis”, the 20th International Conference on Computational Linguistics, COLING, 2004.

[28] R. Harty and C. Ghaddar "Arabic Text Recognition", The International Arab Journal of Information Technology, Vol. 1, No. 2, July. 2004.

[29] S. Connell,” A comparison of hidden markov model features for the recognition of cursive handwriting”, MSc thesis, Michigan State University, 1996.

[30] J. Tohka. Introduction to Pattern Recognition. Institute of Signal Processing Tampere University of Technology, April. 2005.

[31] Bezdek, C. James and S.K. Pal. Fuzzy Models for pattern recognition. IEEE Press, New York, 1992.

[32] R.T. Olszewski, “Generalized feature extraction for structural pattern recognition in time-series data”, Doctor of Philosophy, Carnegie-Mellon Univ, Pittsburgh PA, school of computer science, 2001.

[33] G. A. Abandah, and M. Z. Khedher. "Analysis of handwritten Arabic letters using selected feature extraction techniques", International Journal of Computer Processing of Languages, vol. 22, pp. 49-73, 2009.

[34] A. Nijim, A. El Shenawy, M. T. Mostafa and R. Abo Alez. "A Novel Approach for Recognizing Text in Arabic Ancient Manuscripts", International Journal on Natural Language Computing (IJNLC), Vol. 4, No.6, December 2015.

[35] J.T. Jolliffe. Principal component analysis 2nd. Springer series in statistics, 2002.

[36] k.kim,” face recognition using principal component analysis”, Dcs, University of Maryland, college park, USA, 2003.

[37] P. K. Pandey, Y. Singh and S. Tripathi,“Image Processing using Principle Component Analysis”, International Journal of Computer Applications Volume 15–No.4, pp.0975 – 8887 , February 2011.

[38] K. Sharma, A. Naik and P. Patel, “Study of Artificial Neural Network”, International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) Vol. 2, Issue 4, April 2015.

[39] J.A. Freeman and D.M. Skapura. Neural Networks-Algorithms, Applications, and Programming Techniques. MA, Addison Wesley, Reading, 1991.

[40] D. E. Rumelhart and J.L. McClelland .Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MA, MIT Press, Cambridge, 1986.

[41] H. spatti, A. Flauzino. Artificial Neural Networks: A practical course. Springer, Sep. 2016.

[42] S. Rajasekaran, G.A. Vijayalakshmi. Neural network, fuzzy logic, and genetic algorithm synthesis and application. Synthesis and applications (with cd). PHI Learning Pvt, 2003.

[43] L. I. Perlovsky. Neural Networks and Intellect: Using Model-Based Concepts. Oxford University press, New York, 2001, pp. 63-64.

[44] C.M. BISHOP .Neural Networks for Pattern Recognition. Oxford University Press, UK, 2007.

[45] L. A. Zadeh, ”Fuzzy logic Computer” , Vol. 22, issue 4,1998.

[46] G. Chen and T. Pham. Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems. CRC press, Nov. 2000.

[47] E. Ruspini,”Introduction to Fuzzy set theory basic concepts and structures”, IEEE, 1992.

[48] M. Abdel-Hadi, I. al-Hamdi Al-Hosni, A. Hamid, "Fuzzy logic in the decision-making", Arab Journal of Administrative Sciences, Vol. 6, Kuwait, 1999.

[49] J. Yen and R. Langari, “Fuzzy Logic- Intelligence, Control, and Information”, Pearson Education Pub., First Edition, 2006.

[50] J. Jang,” ANFIS: adaptive-network-based fuzzy inference system“, IEEE Trans. Syst. Man Cybernet. 23, 665–685, 1993.

[51] D. Nauck, F. Klawon; R. Kruse, “Foundations of Neuro-Fuzzy Systems”, J. Wiley & Sons, 1997.

[52] Jang R, “Neuro-Fuzzy Modeling: Architectures, Analyses and Applications”, PhD Thesis, University of California, Berkeley, July 1992.

[53] H. Aljuaid, Z. Muhammad and M. Sarfraz, "A Tool to Develop Arabic Handwriting Recognition System Using Genetic Approach", Journal of Computer Science, vol. 6, 2010.

[54] M. O. Al Barraq and S.C.Mehrotra,"Handwritten Arabic Text Recognition System using Window Based Moment Invariant Method", international journal of advanced research in computer science, Vol. 2, Number 1, Jan-Feb 2011.

[55] R. I. Zaghloul, E. F. Alrawashdeh, D. Mohammad, and K. Bader, " Multilevel Classifier in Recognition of Handwritten Arabic Characters", Journal of Computer Science 7 (4), pp.512-518, 2011.

[56] A. Mesleh1, A. Sharadqh, J. Al-Azzeh, M. Abu-Zaher, N. Al-Zabin, T. Jaber, A.Odeh and M. Hasn, "An Optical Character Recognition", Contemporary Engineering Sciences, Vol. 5, no. 11, pp. 521 – 529, 2012.

[57] X. Wang, T.Huang and X. Liu, "Handwritten Character Recognition Based on BP Neural Network", Third International Conference on Genetic and Evolutionary Computing, 2009.

[58] M. Z. Khedher. G. Al-Talib," Recognition of secondary characters in handwritten Arabic using Fuzzy Logic ", International Conference on Machine Intelligence (ICMI’05), Tozeur, Tunisia, 2005.

[59] M. Sarfraz, S. N. Nawaz and A. Al-Khuraidly," Offline Arabic Text Recognition system" , IEEE International Conference on Geometric Modeling and Graphics, 2003.

[60] S. M. Ismail, S. Norul, and H. Sheikh," Online Arabic Handwritten Character Recognition Based On A rule Based Approach", Journal of Computer Science, pp.1859-1868, 8 September 2012.

[61] M. A. Abdullah, L. M. Al-Harigy, and H. H. Al-Fraidi," Off-Line Arabic Handwriting Character Recognition Using Word Segmentation ", Journal of Computing, Vol. 4, Issue 3, MARCH 2012.

[62] H. Almohri, J. S. Gray and H. Alnajjar, "A Real-time DSP-Based Optical Character Recognition System for Isolated Arabic characters using the TI TMS320C6416T", The IAJC-IJME International Conference, 2008.

[63] L. Cordella, C. D Stefano, F. Fontanella and C. Marrocco, "A Feature Selection Algorithm for Handwritten Character Recognition", Pattern Recognition, IEEE, 19th International Conference, 2008.

[64] G. A. Abandah, K. S. Younis and M. Z. Khedher, "Handwritten Arabic Character Recognition Using Multiple Classifiers Based ON Letter Form", 5th IASTED Int'l Conf. on Signal Processing, Pattern Recognition, & Applications (SPPRA, Innsbruck, Austria, 2008.

[65] F. H. Zawaideh," Arabic Hand Written Character Recognition Using Modified Multi-Neural Network", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, July 2012.

[66] G. R. Ball and S. N. Srihari, "Semi-supervised Learning for Handwriting Recognition", International Conference on Document Analysis and Recognition, IEEE, 2009.

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

[68] S. A. Mahmoud and A. S. Mahmoud, "Arabic Character Recognition using Modified Fourier Spectrum (MFS) ", IEEE, Proceedings of the Geometric Modeling and Imaging, New Trends, 2006.

[69] O. Hachour, "The Combination of Fuzzy Logic and Expert System for Arabic Character Recognition", 3rd International IEEE Conference Intelligent Systems, September 2006.

[70] M. Albakoor, K. Saeed and F. Sukkar, "Intelligent System for Arabic Character Recognition ", IEEE, 2009.

[71] M. Ali, H. Omer and S.L.Ma," Online Arabic Handwriting Character Recognition Using Matching Algorithm", IEEE, 2010.

[72] A.M. ALlMl, "A Neuro-Fuzzy Approach to Recognize Arabic Handwritten Characters", IEEE, 1997.

[73] M. Sarfraz1 and A. T. A. Al-Awami, "Arabic Character Recognition Using Particle Swarm Optimization With Selected And Weighted Moment", IEEE, 2007.

[74] A. Benouareth, A. Ennaji and M. Sellami, "HMMs with Explicit State Duration Applied to Handwritten Arabic Word Recognition", International Conference on Pattern Recognition,", IEEE, 2006.

[75] N. B. Amor and N. E. Ben Amara," Combining a hybrid Approach for Features Selection and Hidden Markov Models in Multifont Arabic Characters Recognition", IEEE Second International Conference on Document Image Analysis for Libraries, 2006.

[76] N. Ben Amor, M. Zarai, and N. E. Ben Amara, "Neuro-Fuzzy approach in the recognition of Arabic Characters ", IEEE, 2006.

[77] A. Zidouri, “PCA-based Arabic Character Feature Extraction ", IEEE, 2007.

[78] H. A. Al Hamad," Use an Efficient Neural Network to Improve the Arabic Handwriting Recognition", International Conference on Systems, Control, Signal Processing and Informatics, 2013.

[79] A. Elbaati, H. Boubaker, M. Kherallah and A. M. Alimi," Arabic Handwriting Recognition Using Restored Stroke Chronology", International Conference on Document Analysis and Recognition 10th, 2009.

[80] A. M. ALIMI, "An Evolutionary Neuro-Fuzzy Approach, to Recognize On-Line Arabic Handwriting”, IEEE, 1997.

[81] M. Namaz, and K. Faez,"Recognition of Multifont Farsi / Arabic Characters Using a Fuzzy Neural Network", IEEE, 1996.

[82] M.T.Sabri and A. Mahmoud, "Offline Arabic Handwritten Text Recognition: A Survey", ACM Computing Surveys, Vol. 45, No. 2, February. 2013.

[83] A. Khawaja, S. Tingzhi, N. M. Memon and A. Rajpa, “Recognition of printed Chinese characters by using Neural Network” , IEEE, pp 169-172, 2006.

[84] K. S. Siddharth, M. Jangid, R. Dhir, R. Rani, “Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features”, International Journal on Computer Science and Engineering, Vol. 3 No. 6 June. 2011.

[85] P. Jhajj and D. Sharma, “Recognition of Isolated Handwritten Characters in Gurmukhi Script”, International Journal of Computer Applications, Vol. 4, No. 8, 2010.

[86] M. Kef1, L.Chergui and S. Chikhi, “A novel fuzzy approach for handwritten Arabic character recognition”, Springer-Verlag London 2015.

[87] G. A. Montazer and H. Q. Saremi,”A neuro-fuzzy inference engine for Farsi numeral characters recognition”, Expert Systems with Applications 37, 6327–6337, 2010.

[88] Ch. Leila, K. Maâmar and Ch. Salim, “Combining Neural Networks for Arabic Handwriting Recognition”, IEEE 978-1-4577-0908-1/11, 2011.

[89] R. I. M. Elanwar, M. A. A. Rashwan and S. Mashali, “A Multiple Classifiers System For Solving The Character Recognition Problem In Arabic Alphabet”, Conference Paper, December 2006.

[90] B. Alijla and K. Kwaik, “OIAHCR: Online Isolated Arabic Handwritten Character Recognition Using Neural Network”, the International Arab Journal of Information Technology, Vol. 9, No. 4, Jul. 2012.

[91] R.Al-Jawfi, “Off Handwriting Arabic Character Recognition LeNet Using Neural Network”, the International Arab Journal of Information Technology, vol. 6, no. 3, pp. 304-309, 2009.

[92] A. Alaei, P. Nagabhushan and U. a Pal, “A New Two-stage Scheme for the Recognition of Persian Handwritten Characters”, 12th International Conference on Frontiers in Handwriting Recognition, IEEE, 2010.

[93] A. Shaout and O. Balola ,"Isolated Arabic Handwritten Character Recognition: Survey", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, October 2014.

[94] B. Anuradha, B. Koteswarrao, "An efficient Binarization technique for old documents", International conference on Systemic, Cybernetics, and Informatics (ICSCI2006), Hyderabad, 2006, pp771-775.

[95] A. Rosenberg and N. Dershowitz, "Using SIFT Descriptors for OCR of Printed Arabic", Tel Aviv University, 2012.

[96] M. Dahi, N. A. Semary and M. M. Hadhoud, "Primitive Printed Arabic Optical Character Recognition using Statistical Features", IEEE Seventh International Conference on Intelligent Computing and Information, Systems. 2015.

[97] R. Sharma, M.S. Patterh "A new pose invariant face recognition system using PCA and ANFIS", Elsevier GmbH, 2015.

[98] Fi-John Chang, Ya-Ting Chang, "Adaptive neuro-fuzzy inference system for prediction of water level in reservoir", National Taiwan University, Taiwan, pp.9 - 1–10, 2006.

[99] R. Kohavi, and F. Provost, "Glossary of Terms Special Issue on Applications of Machine Learning and the Knowledge Discovery Process", Machine Learning **30**, pp. 271-274, 1998.

[100] Y. Cheng, Felix X. Yu, R. S. Feris and A. Choudhary, "An exploration of parameter redundancy in deep networks with circulant projections", Proceedings of the IEEE International Conference on Computer Vision, 2015.

[101] O. Balola , A. Shaout, M. Elhafiz, "Two stage classifier for Arabic Handwritten Character Recognition", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 12, December 2015.

Appendix A: Off-line Isolated Arabic Handwritten Alphabet Data set

This appendix contents the some samples of forms that filled by student to structure the Data set of SUST-ARG.

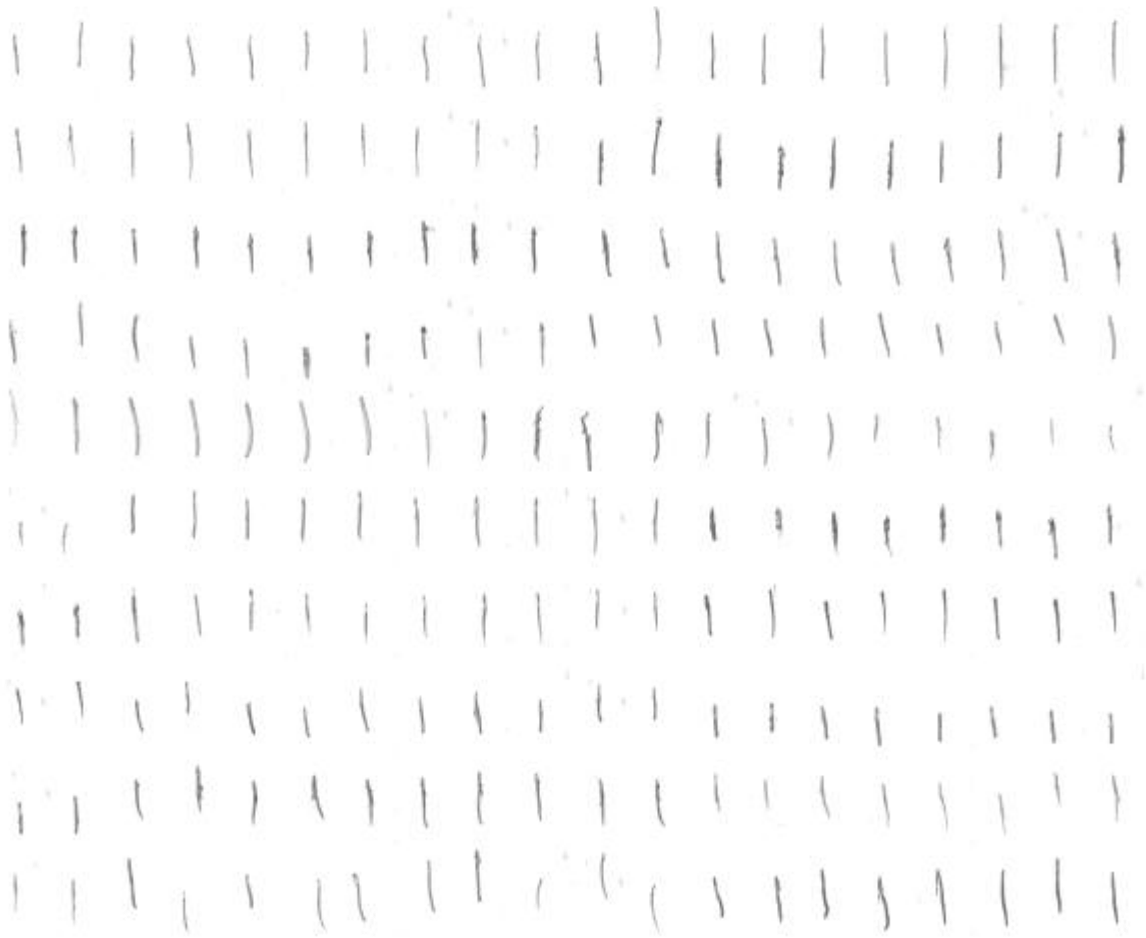


Figure A.1 Character (ا)

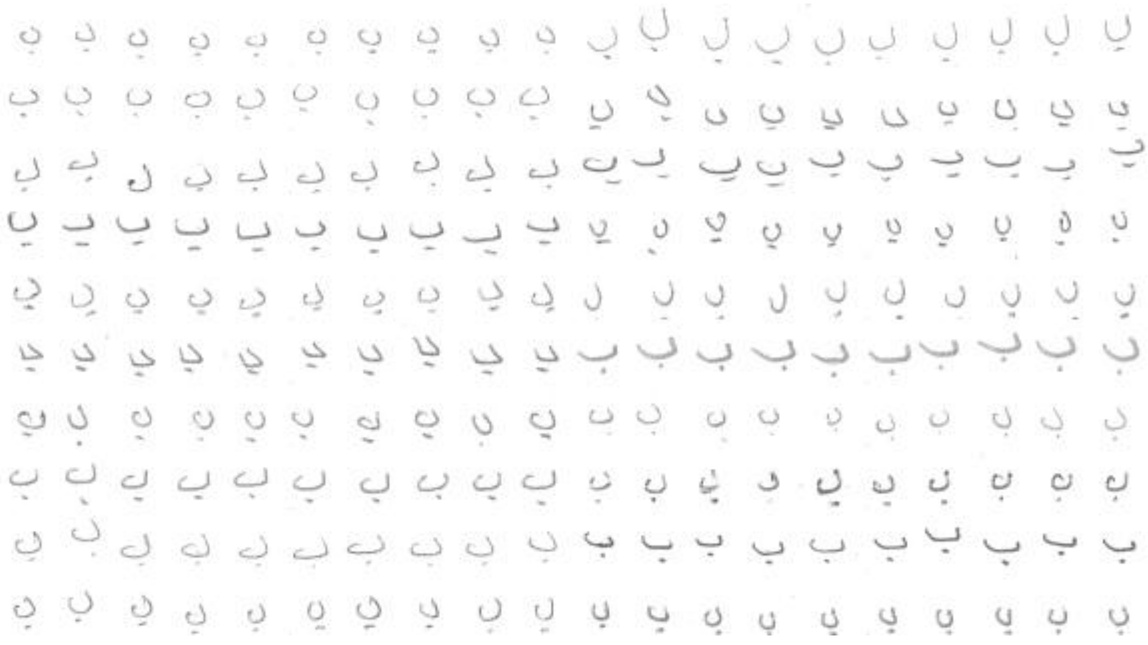


Figure A.2 Character (ب)

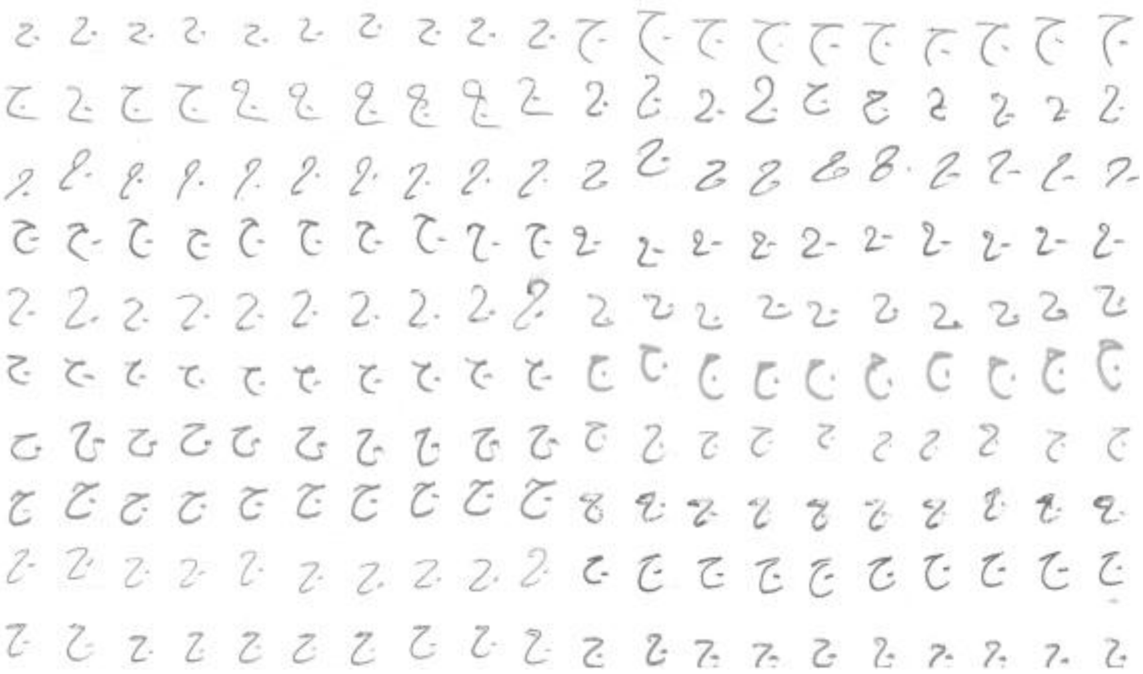


Figure A.3 Character (ج)

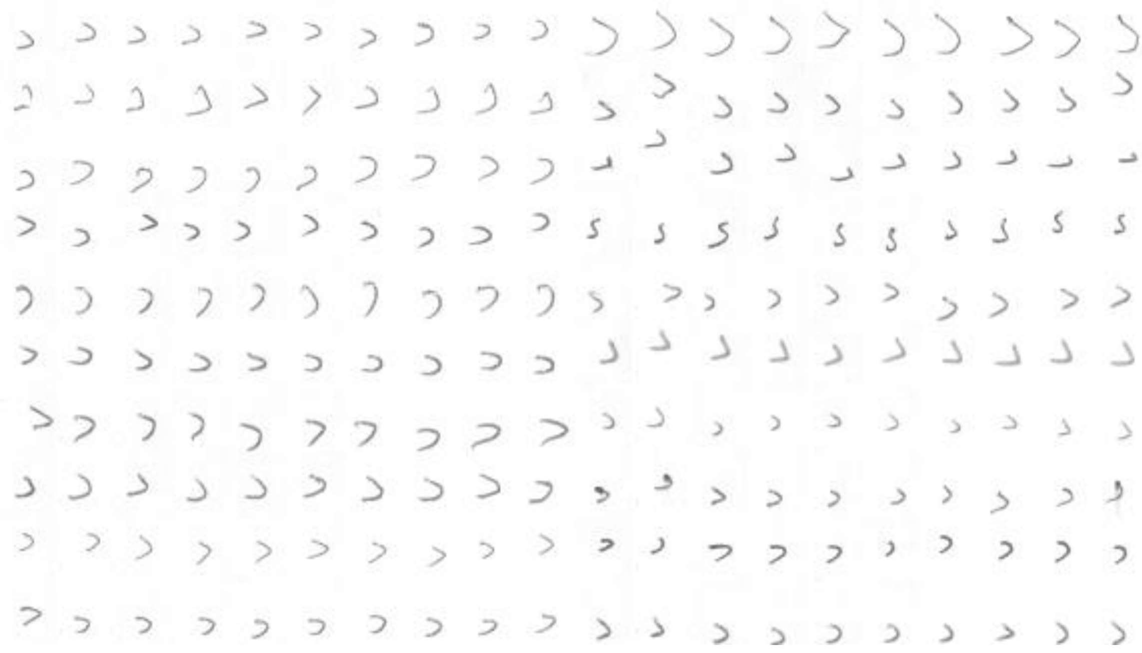


Figure A.4 Character (د)

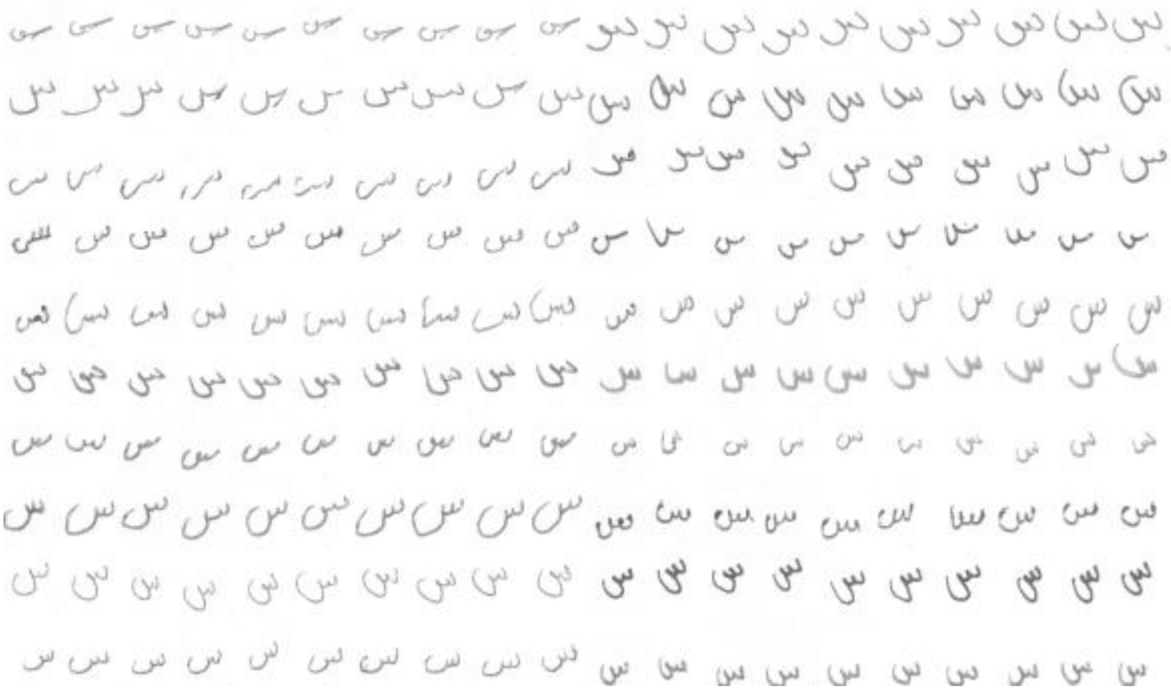


Figure A.5 Character (س)

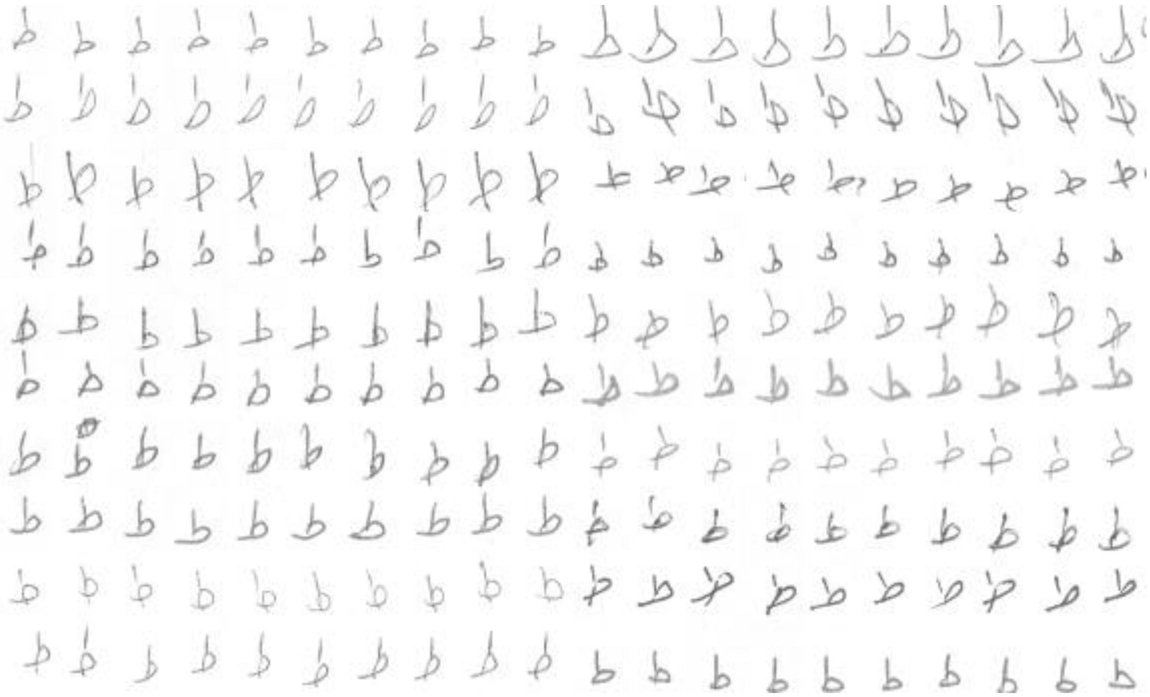


Figure A.6 Character (ط)

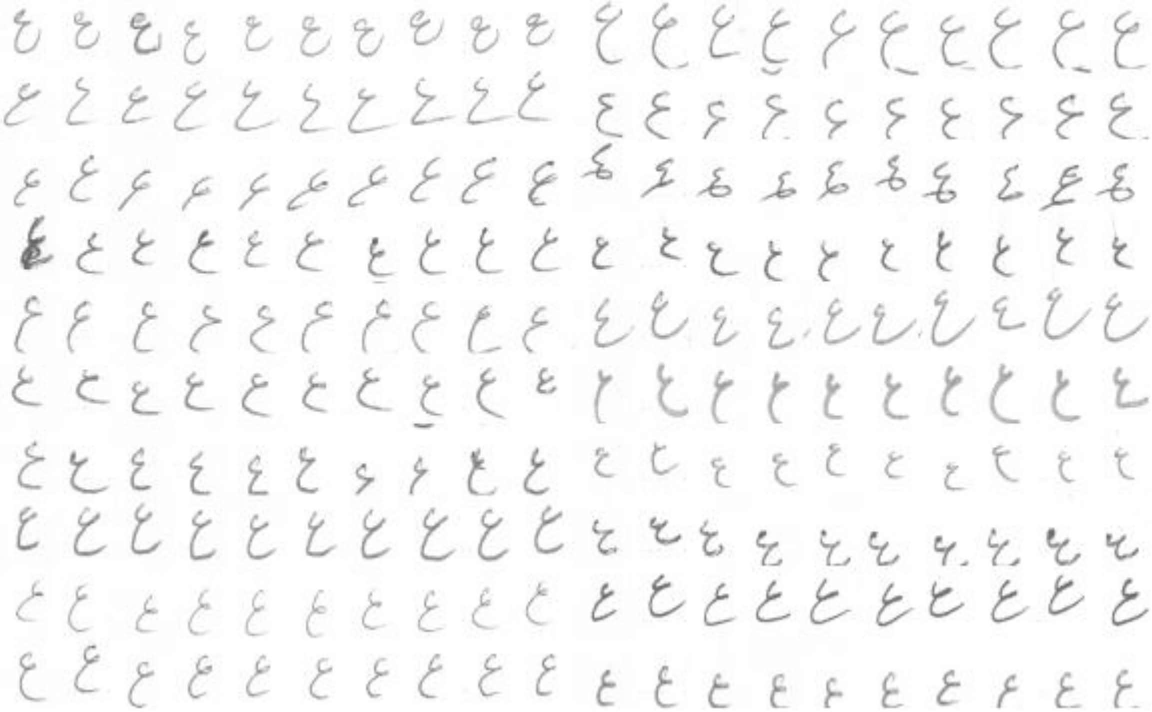


Figure A.7 Character (ع)

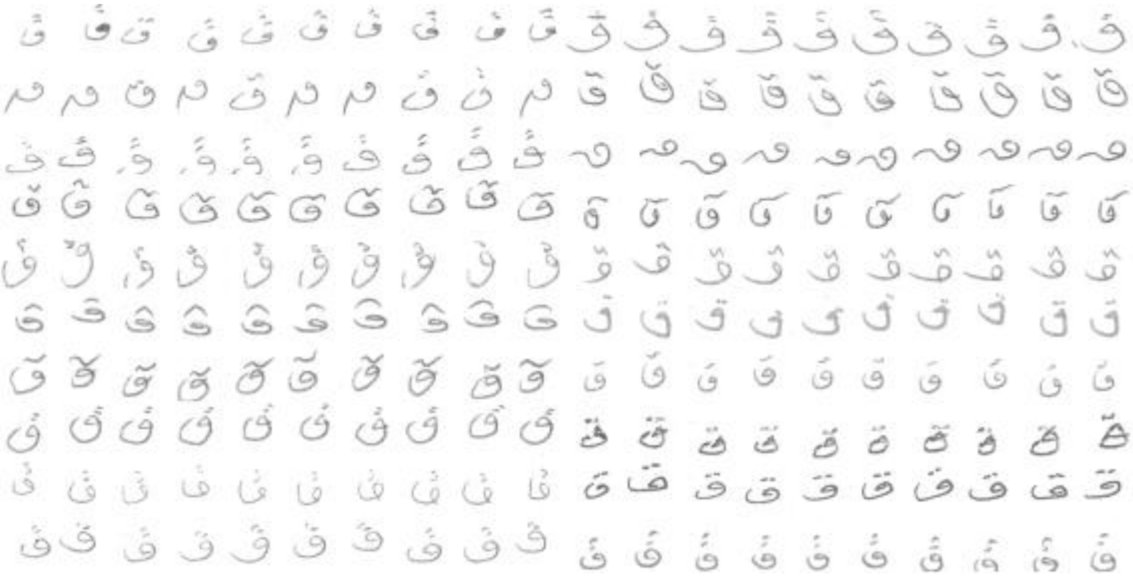


Figure A.8 Character (ق)

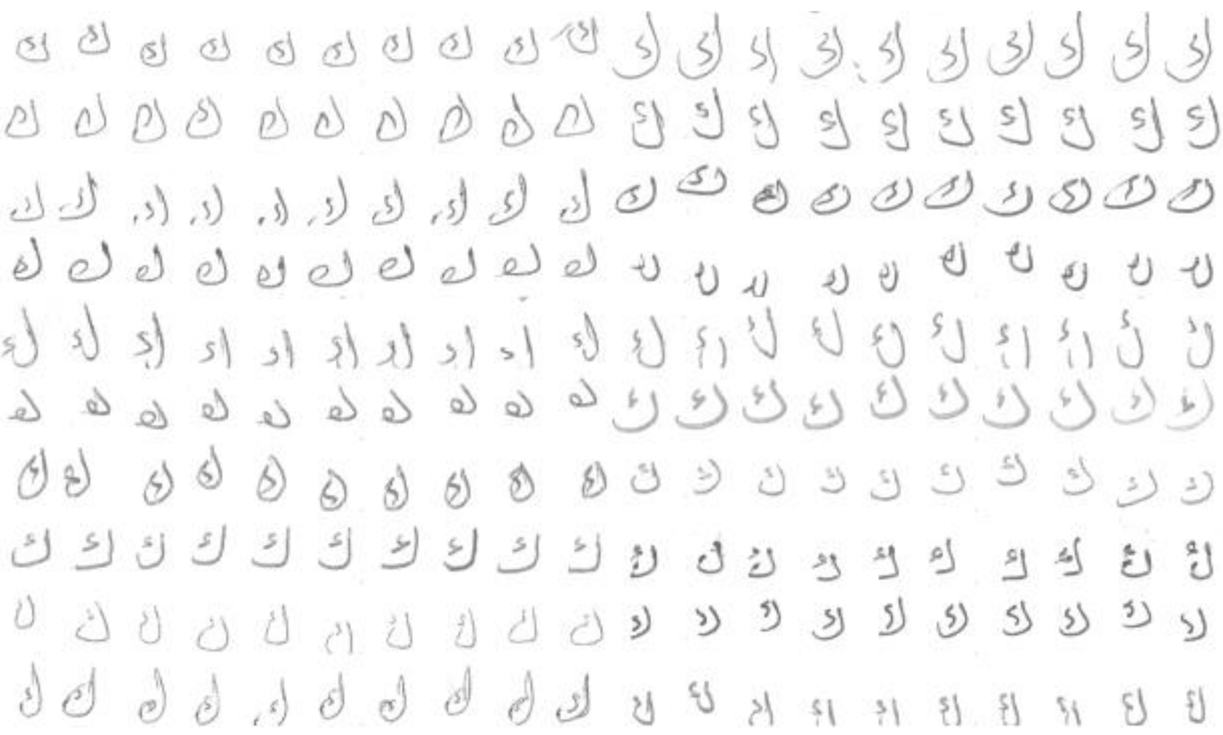
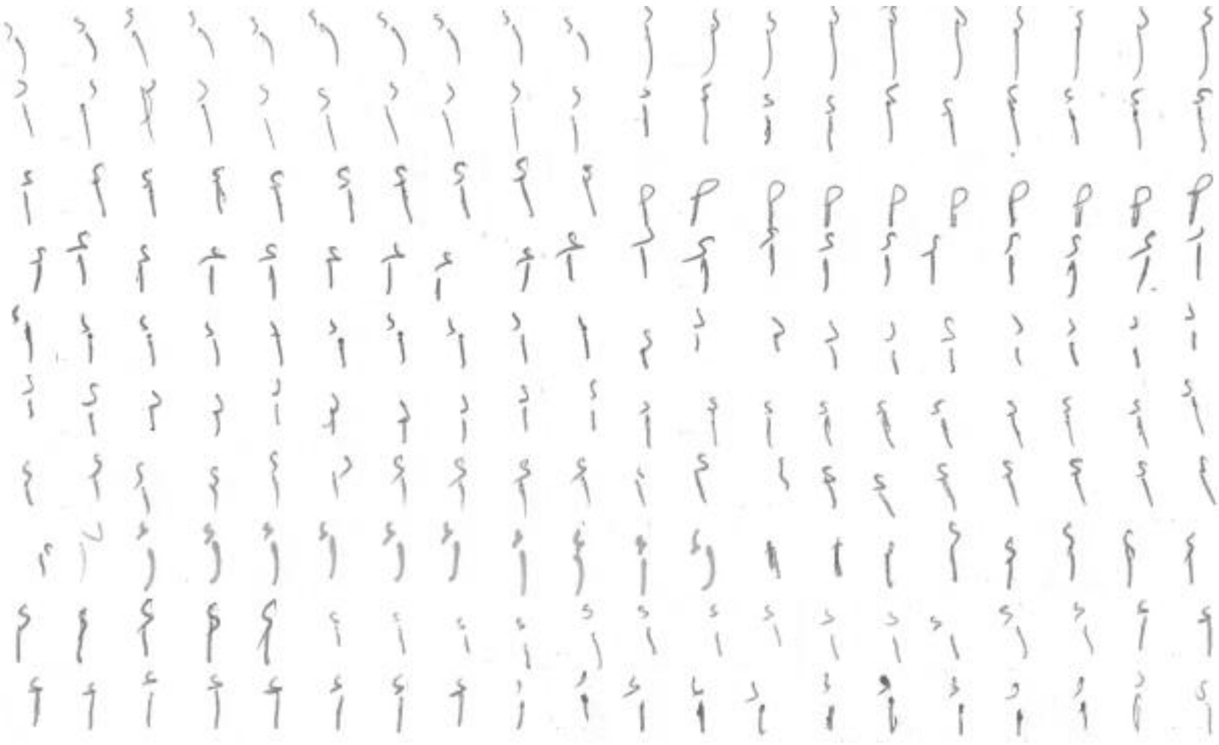


Figure A.9 Character (ك)



FigureA.10 Character (أ)

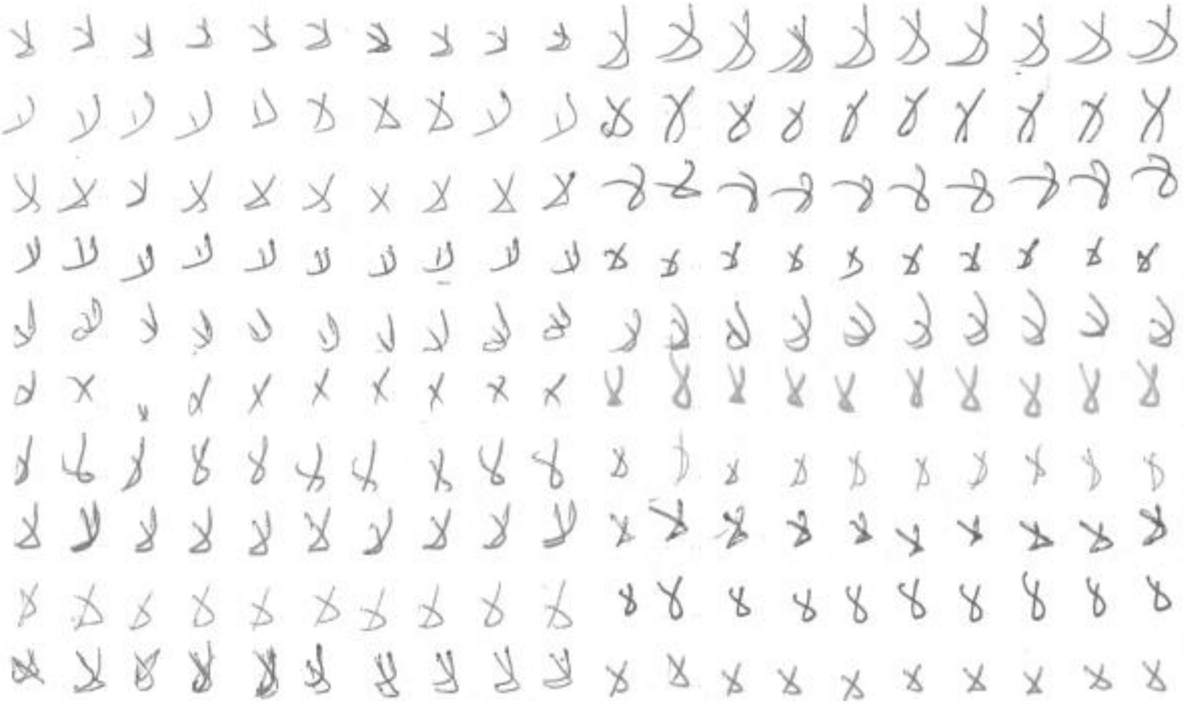
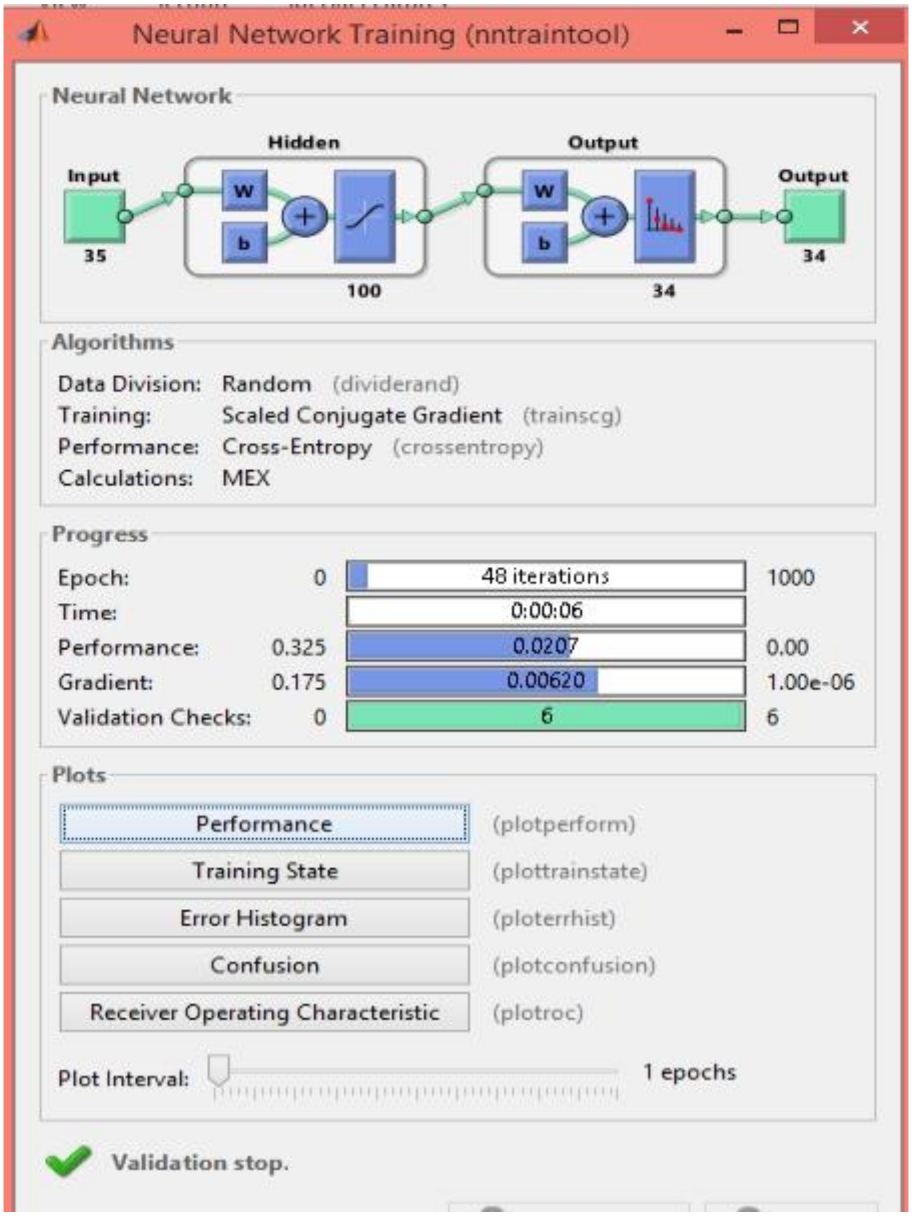


Figure A.11 Character (لا)

Appendix B: Matlab GUI for the System Processing

This appendix explains the screens of ANN training tools and screens for ANFIS structure rules, editor and rules viewer.



Algorithms

Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Cross-Entropy (crossentropy)
Calculations: MEX

Progress

Epoch:	0	48 iterations	1000
Time:		0:00:06	
Performance:	0.325	0.0207	0.00
Gradient:	0.175	0.00620	1.00e-06
Validation Checks:	0	6	6

(plotperform)
(plottrainstate)
(ploterrhist)
(plotconfusion)
(plotroc)

Plot Interval: 1 epochs

✓ Validation stop.

Figure B.1 Training and testing the network Toll

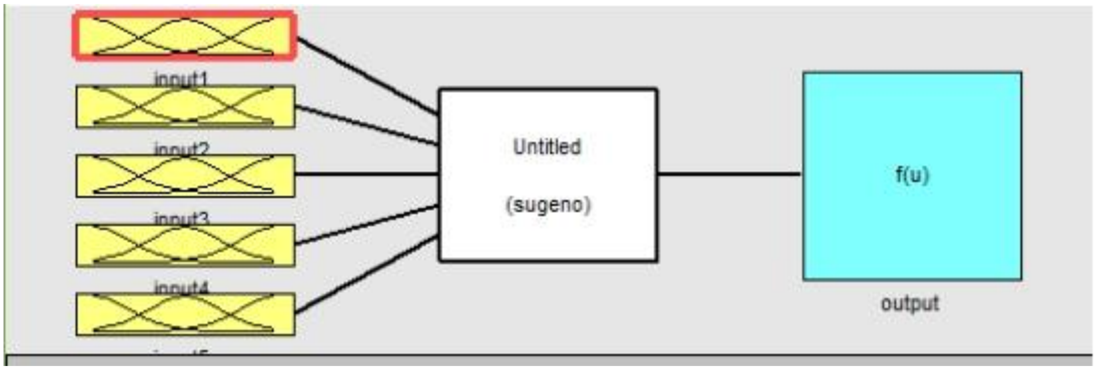
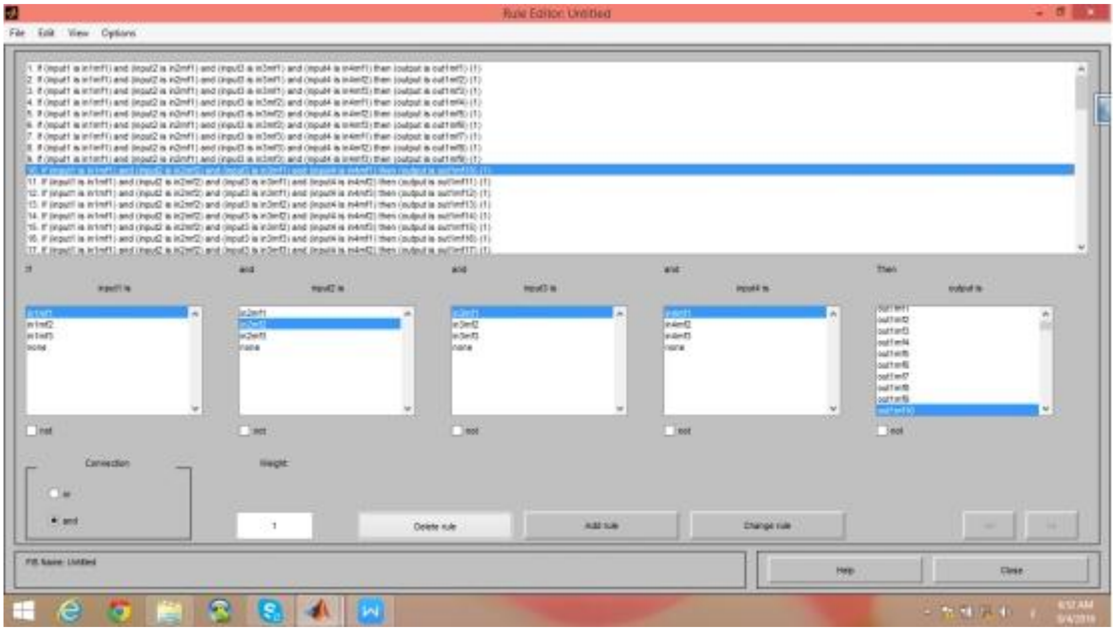


Figure B.2 FIS editor, Sugeno system with 5 inputs



1. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) then (output is out1mf1) (1)
2. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) then (output is out1mf2) (1)
3. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf3) then (output is out1mf3) (1)
4. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) and (input4 is in4mf1) then (output is out1mf4) (1)
5. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) and (input4 is in4mf2) then (output is out1mf5) (1)
6. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf2) and (input4 is in4mf3) then (output is out1mf6) (1)
7. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf3) and (input4 is in4mf1) then (output is out1mf7) (1)
8. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf3) and (input4 is in4mf2) then (output is out1mf8) (1)
9. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf3) and (input4 is in4mf3) then (output is out1mf9) (1)
10. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf1) and (input4 is in4mf1) then (output is out1mf10) (1)
11. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf1) and (input4 is in4mf2) then (output is out1mf11) (1)
12. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf1) and (input4 is in4mf3) then (output is out1mf12) (1)
13. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf2) and (input4 is in4mf1) then (output is out1mf13) (1)
14. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf2) and (input4 is in4mf2) then (output is out1mf14) (1)
15. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf2) and (input4 is in4mf3) then (output is out1mf15) (1)
16. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf3) and (input4 is in4mf1) then (output is out1mf16) (1)
17. If (input1 is in1mf1) and (input2 is in2mf2) and (input3 is in3mf3) and (input4 is in4mf2) then (output is out1mf17) (1)

Figure B.3 Rule editor

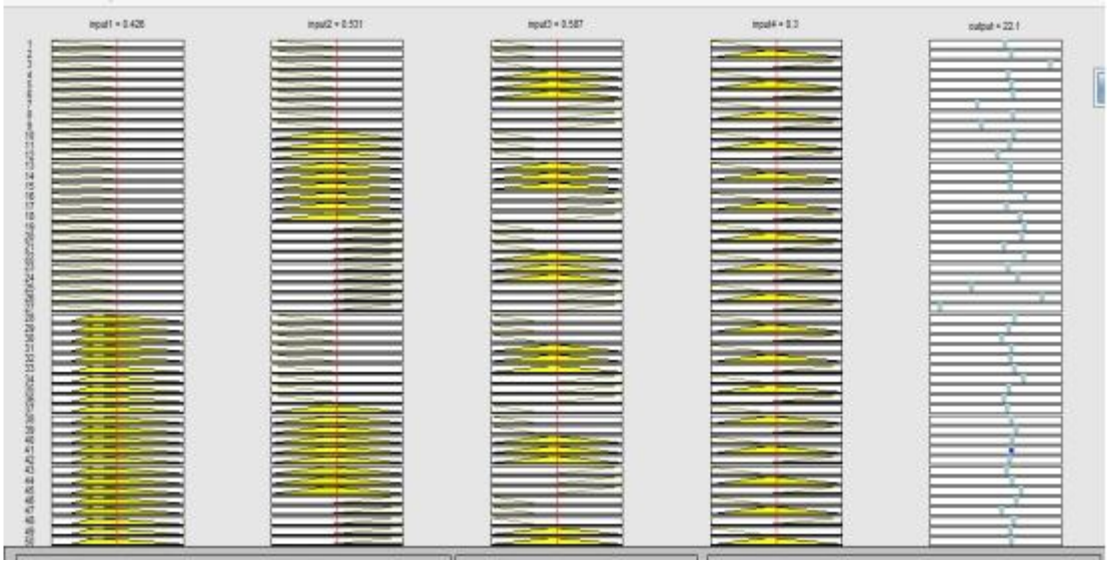
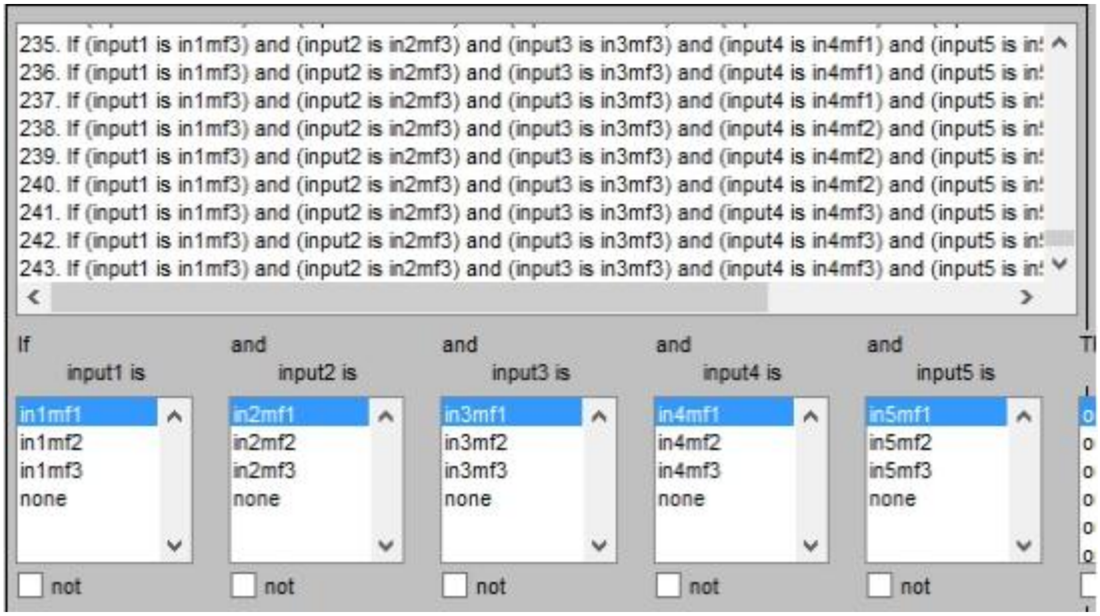


Figure B.4 Rule viewer



```
235. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf1) and (input5 is in!
236. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf1) and (input5 is in!
237. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf1) and (input5 is in!
238. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf2) and (input5 is in!
239. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf2) and (input5 is in!
240. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf2) and (input5 is in!
241. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf3) and (input5 is in!
242. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf3) and (input5 is in!
243. If (input1 is in1mf3) and (input2 is in2mf3) and (input3 is in3mf3) and (input4 is in4mf3) and (input5 is in!
```

If input1 is	and input2 is	and input3 is	and input4 is	and input5 is	Th
in1mf1	in2mf1	in3mf1	in4mf1	in5mf1	o
in1mf2	in2mf2	in3mf2	in4mf2	in5mf2	o
in1mf3	in2mf3	in3mf3	in4mf3	in5mf3	o
none	none	none	none	none	o
					o
					o
□ not	□ not	□ not	□ not	□ not	□

Figure B.6 Rules with 5 inputs

Appendix C: Source Code

This appendix described the system code preprocessing functions and features extraction techniques function and ANFIC classifier.

Appendix C1: Preprocessing

```
% Preprocessing includes binarization, cropping, resize, morphology and
normalization

% Find the boundary of the image, inputs gray scale images and outputs
the matrix of numbers
function bw2 = edu_imgcrop(bw)
%bw =load[ 'D:\PROGRESS REPORT\dataset_group15\train\L2\f2letter41.bmp'];
[y2temp x2temp] = size(bw);
x1=1;
y1=1;
x2=x2temp;
y2=y2temp;
% Finding left side blank spaces
cntB=1;
while (sum(bw(:,cntB))==y2temp)
    x1=x1+1;
    cntB=cntB+1;
end
% Finding right side blank spaces
cntB=1;
while (sum(bw(cntB,:))==x2temp)
    y1=y1+1;
    cntB=cntB+1;
end

% Finding upper side blank spaces
cntB=x2temp;
while (sum(bw(:,cntB))==y2temp)
    x2=x2-1;
    cntB=cntB-1;
end
```

```
% Finding lower side blank spaces
cntB=y2temp;
while (sum(bw(cntB,:))==x2temp)
    y2=y2-1;
    cntB=cntB-1;
end

% Crop the image to the edge
bw2=imcrop(bw,[x1,y1,(x2-x1),(y2-y1)]);
```

```
% This function will take the cropped binary image and change it to 5 x 7
the inputs are binary numbers and outputs is matrix of 7 by 5

function lett = f_e(bw2)

bw_7050=imresize(bw2,[70,50]);
%imshow(bw_7050)
for cnt=1:7
    for cnt2=1:5
        Atemp=sum(bw_7050((cnt*10-9:cnt*10),(cnt2*10-9:cnt2*10)));
        lett((cnt-1)*5+cnt2)=sum(Atemp);
    end
end

lett=((100-lett)/100);
```

Appendix C2: Feature Extraction

CCOB Features

Center of mass

% calculate the the center of image

```
function cent=get_CenterOfMass_7(bw)
% figure,imshow(bw)
s = regionprops(bw, 'centroid');
centroids = cat(1, s.Centroid);
```

```
cent=[centroids(1,2) centroids(1,1)];
```

---

black in histogram

```
function feature=get_BlackIncHisto_7(bw,f_dim)
% clear
% [name,path]= uigetfile('*.*');
% image =imread([path,name]);
% bw=~im2bw(image,0.9);
% f_dim=20;
[height,width]=size(bw);
vertical=sum(bw);
vertical(find(vertical==0))=[];
max_v=max(vertical);
vertical=vertical/max_v;
horizontal=sum(bw,2);
horizontal(find(horizontal==0))=[];
max_h=max(horizontal);
horizontal=horizontal'/max_h;
length_vertical=length(vertical);
length_horizontal=length(horizontal);
for i=1 :f_dim
    index_vertical=round(i*length_vertical/f_dim);
    index_horizontal=round(i*length_horizontal/f_dim);
    if(index_vertical==0)
        index_vertical=1;
    end
    if(index_horizontal==0)
        index_horizontal=1;
    end
    v_hist(i)=vertical(index_vertical);
    h_hist(i)=horizontal(index_horizontal);
end
feature=[v_hist/h_hist];
%feature=[v_hist';h_hist'];
d=[];
```

---

cross of mass function

```
function cross=get_Cross_7(bw)
bw2 = bwmorph(bw, 'shrink',Inf);
% imshow(bw3)
[row,col] = find(bw2);
row=mean(row);
col=mean(col);
cross=[row,col];
```

Appendix C3: Classification

Reduce features using PCA

```
close all;clear;clc;
myfolder = 'D:\PROGRESS REPORT\DATASET_GROUPS\GROUP14\L';
features_main=[];
features_t=[];
img_num_traing=150;
c=0;
d=0;
num_eigenfaces=5;
for m=1:1:2
    my_folder=strcat(myfolder,num2str(m));
    filenames = dir(fullfile(my_folder, '*.bmp')); %contains
properties(not content) of all files
    total_images = numel(filenames);%total number of png flies in folder
    for n = 1:150 %THIS LOOP READS ALL IMAGES
        full_name= fullfile(my_folder, filenames(n).name);%e.g a string
of name 'image123.png'
        imgGray=imread(full_name);
        bw = im2bw(imgGray,graythresh(imgGray));
        bw2 = edu_imgcrop_1(bw);
        bw_img=edu_imgresize_1(bw2);
        d=d+1;
        features_main(:,d)=bw_img(:);  %apply PCA here
    end
    for n = 151:1:total_images %THIS LOOP READS ALL IMAGES
        full_name= fullfile(my_folder, filenames(n).name);%e.g a string
of name 'image123.png'
        imgGray=imread(full_name);
        bw = im2bw(imgGray,graythresh(imgGray));
        bw2 = edu_imgcrop(bw);
        bw2_img=edu_imgresize(bw2);
        c=c+1;
        features_t(:,c)=bw2_img(:);   %apply PCa here
    end

end
mean_v=mean(features_main,2);
train_shiftedimgs=features_main-repmat(mean_v,1,size(features_main,2));
[evectors,evalues,A]=princomp(features_main');
evectors=evectors(:,1:num_eigenfaces);
traning_features=evectors'*train_shiftedimgs;
test_shiftedimgs=features_t-repmat(mean_v,1,size(features_t,2));
testing_features=evectors'*test_shiftedimgs;

T=[];
for o=1:2
    T=[T o*ones(1,150)];
end
training_inputs=[traning_features',T']
T_test=[];
for o=1:2
    T_test=[T_test o*ones(1,50)];
end
testing_input=[testing_features',T_test'];
save(train_name,'training_inputs');
save(test_name,'testing_input');
```

Anfis parameters

```
 numMFs=3;
 mfType='trimf';
%Then FIS-matrix fismat is generated by command genfish1:
 fismat=genfis1(training_inputs,numMFs,mfType);
 [x,mf]=plotmf(fismat,'input',1);
 plot(x,mf);
 numEpochs=1000;
```

```
out=0;
[fismat1,trnErr,ss,fismat2,chkErr]=anfis(training_inputs,fismat,numEpochs
,NaN,testing_input);
trnOut=evalfis(traning_features,fismat1);
 trnRMSE=sqrt(sum(trnOut-training_inputs(:,2))/length(trnOut))
 chkOut=evalfis(testing_features,fismat2);
 chkRMSE=sqrt(sum(chkOut-testing_input(:,2))/length(chkOut))
 %sqrt(sum((targetdata-testOut)'*(targetdata-
testOut))/length(targetdata));
 save(train_name,'trnOut','trnRMSE','out');
  save(test_name,'chkOut','chkRMSE');
 epoch=1:numEpochs;
 plot(epoch,trnRMSE,'O',epoch,chkRMSE,'x')
 hold on;
 hold on;
% plot(epoch,[trnRMSE chkRMSE])
 hold off;
```

ANN classifier: training, testing, confusion

```
close all;clear;clc;

OCRTrainingFileName1='D:\PROGRESS REPORT\dataset_200\300 dataset\PCA\25
features\training_result';
OCRTestingFileName1='D:\PROGRESS REPORT\dataset_200\300 dataset\PCA\25
features\testing_result';
OCRconfusionFileName= 'D:\PROGRESS REPORT\dataset_200\300 dataset\PCA\25
features\confusion_result';
OCRconfusiontrFileName='D:\PROGRESS REPORT\dataset_200\300 dataset\PCA\25
features\confusion_result';
%myfolder = strcat(cd,'\AHCR_DATASET\L');
myfolder = 'D:\DATASET_PROCESED\NEW DATASET_700_200\testing\L';
c=0;
d=0;
%myfolder = strcat(cd,'\AHCR_DATASET\L');
features_main=[];
features_t=[];
img_num_traing=200;
```

```
num_eigenfaces=25;
for m=1:1:34
    my_folder=strcat(myfolder,num2str(m));
    filenames = dir(fullfile(my_folder, '*.bmp')); %contains
properties(not content) of all files
    total_images = numel(filenames);%total number of png flies in folder
    for n = 1:1:200 %THIS LOOP READS ALL IMAGES
        full_name= fullfile(my_folder, filenames(n).name);%e.g a string
of name 'image123.png'
        imgGray=imread(full_name);
        bw = im2bw(imgGray,graythresh(imgGray));
        bw2 = edu_imgcrop(bw);
        bw_img=edu_imgresize(bw2);
        d=d+1;
        features_main(:,d)=bw_img(:);
    end

    for n = 201:1:total_images %THIS LOOP READS ALL IMAGES
        full_name= fullfile(my_folder, filenames(n).name);%e.g a string
of name 'image123.png'
        imgGray=imread(full_name);
        bw = im2bw(imgGray,graythresh(imgGray));
        bw2 = edu_imgcrop(bw);
        bw2_img=edu_imgresize(bw2);
        c=c+1;
        features_t(:,c)=bw2_img(:);
    end
end
T=[];
for o=1:34
    T=[T o*ones(1,200)];
end
mean_v=mean(features_main,2);
train_shiftedimgs=features_main-repmat(mean_v,1,size(features_main,2));
[evectors,evalues,A]=princomp(features_main');
evectors=evectors(:,1:num_eigenfaces);
traning_features=evectors'*train_shiftedimgs;
test_shiftedimgs=features_t-repmat(mean_v,1,size(features_t,2));
```

```
testing_features=evectors'*test_shiftedimgs;

a_features_input_training=traning_features;
a_features_target=T;
a_features_input_testing=testing_features;
x = a_features_input_training;
t = full(ind2vec(a_features_target))  ;
% Choose a Training Function
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainscg';  % Scaled conjugate gradient backpropagation.
% Create a Pattern Recognition Network
hiddenLayerSize = 100;
net = patternnet(hiddenLayerSize);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 90/100;
net.divideParam.valRatio = 10/100;
%net.divideParam.testRatio = 20/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x)
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
m_nn=[tind;yind];
pe = sum(tind ~= yind)/numel(tind);
trainAccuracy=  num2str((1-pe)*100);
display(['Accuracy trained data=' num2str((1-pe)*100)]);
y2 = eye(34);
y1=[];
for i=1:34
for j=1:200
```

```
y1=[y1;y2(i,:)];
end;
end;
y2=y1';
[c,cm,ind,per] = confusion(y2,y);
%OCRTestingFileName = strcat(int2str(i),'_confusion_1');
save(OCRconfusiontrFileName,'c','cm','per');
%save training
save(OCRTrainingFileName1,'yind','percentErrors','trainAccuracy');
% View the Network
%view(net)
 y_tst=net(a_features_input_testing);
 yind_tst = vec2ind(y_tst);
 res=[];
for o=1:34
    res=[res o*ones(1,100)];
end
r=[res;yind_tst];
t_ind_tst=[];
 for o=1:34
     t_ind_tst=[t_ind_tst o*ones(1,100)];
 end
percentErrors = sum(t_ind_tst ~= yind_tst)/numel(t_ind_tst)
display(['Accuracy tested data=' num2str((1-percentErrors)*100)]);
testAccuracy=num2str((1-percentErrors)*100);
%save files test
save(OCRTestingFileName1,'y_tst','percentErrors','testAccuracy');
y = eye(34);
y1=[];
for i=1:34
for j=1:100
y1=[y1;y(i,:)];
end;
end;
y=y1';
[c,cm,ind,per] = confusion(y,y_tst);
%OCRTestingFileName = strcat(int2str(i),'_confusion_1');
save(OCRconfusionFileName,'c','cm','per');
```

LIST OF PUBLICATIONS

[1] Omar Balola Ali and Adnan Shaout, “Isolated Arabic Handwritten Character Recognition: A Survey”, International Journal of Advanced Research in Computer Science and Software Engineering **(IJARCSSE),** Vol. 4, Issue 10, October 2014.

[2] Omar Balola, Adnan Shaout and Mohammed Elhafiz, “Two stage classifier for Arabic Handwritten Character Recognition”, International Journal of Advanced Research in Computer and Communication (***IJARCCE),*** Vol. 4, Issue 12, December 2015.

[4] Omar Balola Ali and Adnan Shaout, “Hybrid Arabic Handwritten Character Recognition Using PCA and ANFIS”, The International Arab Conference on Information Technology (ACIT’2016), Morocco, 6-8 December 2016.

[4] Omar Balola Ali and Adnan Shaout, “A Hybrid Recognition System for Islamic Annotation and Historical Arabic Handwritten Manuscripts”, 4th International Conference on Islamic Applications in Computer Science and Technology, Sudan, 20-22 Dec 2016.