*Republic of Iraq*
*Ministry of Higher Education*
*and Scientific Research*
*Al-Nahrain University*
*College of Science*

# Arabic Character Recognition Based on Moments Method

**A Thesis**
**Submitted to the College of Science, Al-Nahrain University as a Partial Fulfillment of the Requirements for the Degree of Master of Science in**
**Computer science**

**By**

**Raghad Khrebit Rashid Al-Khalidy**

**(B. Sc. 2005)**

**Supervisors**

**Dr. Laith Abdul Aziz Al-Ani** **Dr. Taha S. Bashaga**

**2008** **1429**

# بسم الله الرحمن الرحيم

((هذا من فضل ربي ليبلوني ء اشكر أم اكفر ومن شكر فانما يشكر لنفسه ومن كفر فان ربي غني كريم))

**صدق الله العلي العظيم**

القرآن الكريم – سورة النمل ، أية ٤٠

# الاهداء

الى العراق........وطني الاوحد.
الى كل من ساهم بشرف في بنائه.
الى من ربياني صغيرا ورعياني كبيرا
ابي ....شجرة
السنديان الشامخة
امي....زهرة
النرجس العطرة

عائلتي..............

اهدي ثمرة جهدي المتواضع

رغد

# Acknowledgment

**First of all, my great thank to Allah who helped me and give me the ability to achieve this work.**

**My great thanks to my supervisors Dr. Laith Abdul Aziz AL-Ani and Dr. Taha S. Bashaga for their guidance, supervision and efforts during this work.**

**My deep gratitude to the employees and stuff of the Computer Science Department.**

**Thanks to my parents, sisters and my brothers for their help and patience during this work.**

**I also produce my gratitude to Dr. Mohamed Sahib Al-Taee and Dr. Mustafa Dhiaa Al-Hassani for their help and cooperation.**

**Finally thanks to all my friends for Supporting and giving me advices.**

**Raghad**

## Supervisor Certification

We certify that this thesis was prepared under our supervision at the Department of Computer Science /College of Science /Al- Nahrain University, by **Raghad Khrebit Rashid Al-Khalidy** as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:

Name: **Dr.Laith Abdul Aziz AL-Ani**

Title: **Assist.Prof**

Date: / / 2008

Signature:

Name: **Dr.Taha S. Bashaga**

Title**: Lecturer**

Date: / / 2008

In view of the available recommendations, we forward this thesis for debate by the examination committee.

Signature:

Name: **Dr.Taha S. Bashaga**

Title: **Head of the Department of Computer Science,**

**Al-Nahrain University.**

Date: / / 2008

# *Certification of the Examination Committee*

We certify that we have read this thesis and as an examining committee, examined the student in its content and what is related to it and that in our opinion it meets the standard of a thesis for the degree of Master of Science in Computer Science.

Signature:
Name: **Dr.Abdul Monem S.Rahma**
Title: **Assist.Prof**
Date: / **/** 2009
**(chairman)**

Signature:
Name: **Dr. Ban N. Dhannoon**
Title: **Assist.Prof**
Date: / / 2009
**(Member)**

Signature:
Name: **Dr. Sawsan Kamal Thamer**
Title: **Lecturer**
Date: / / 2009
**(Member)**

Signature:
Name:**Dr. Laith Abdul Asis Al-Ani**
Title: **Assist.Prof**
Date: / / 2009
**(supervisor)**

Signature:
Name: **Dr. Taha S. Bashaga**
Title: **Lecturer**
Date: / / 2009
**(supervisor)**

Approved by the dean of the collage of science, Al-Nahrain University.

Signature:
Name: **Dr.Laith Abdul Aziz AL-Ani**
Title: **Assistant Profesor**
Date: **/** **/** 2009
**(Dean of Collage of Science)**

# Abstract

Pattern recognition is an essential part of any high-level image analysis systems.

Arabic language has four forms for each letter depending on the position of the letter in each word. These are initial, medial, final and isolated. This research concerned with recognizing an isolated Arabic Characters using Hu's seven moments concept.

Moments and functions of moments have been extensively employed as invariant global features of images in characters recognition.

Character recognition performed by two stages; first is the training stage and the second is testing stage, these involves several major functions starting from the input character until deciding the recognition of character.

These functions are preprocessing, feature extraction and characters matching. Preprocessing function includes; image acquisition, noise removal using median filter, image binarization and characters segmentation.

The characters matching include the computation of the Euclidean distance between testing and training characters.

Recognition system has been performed on the printed Arabic characters and the percentage accuracy for recognize 65 printed characters was 96.92307% While in hand written characters the recognition percentage is decreased, due to irregularities appears in the handwritten characters.

# Table of Contents

# Table of Contents

# *List of Abbreviations*

| | |
|---|---|
| **RMS** | **Root Mean Square** |
| **OCR** | **Optical Character Recognition** |
| **DCG** | **Definite Clause Grammar** |
| **ACR** | **Automatic Character Recognition** |
| **DPI** | **Dot Per Inch** |
| **BPP** | **Bits Per Pixel** |

# Chapter One

# General Introduction

# Chapter One
# General Introduction

## 1.1 Introduction

One of the most frequent tasks in computer vision and image processing is the recognition of an image or an object in the image. Among these tasks, Optical Character Recognition (OCR).

Various OCR algorithms have been found to achieve better performances. These algorithms include template matching, image signatures, image geometric features, and shape-based image invariants. Among these algorithms, the shape-based image invariants are of particular interests as they have the invariance property which can improve the recognition performance even the images have undergone various image transformations such as translation, scaling and rotation. There are two types of shape-based image invariants: boundary-based and region-based **[Zha02].** The boundary-based image invariants focus on the properties contained in the image's contour. The most common boundary-based image invariants include Fourier descriptors. The region-based image invariants take the whole image area as the research object. Region-based image invariants include various moment-based invariants such as Hu's seven moment invariants.

The aim of the present work is to make computers able to recognize optical symbols without human intervention. This is accomplished by searching a match between the features extracted from the given symbol's image and the library of image models. Ideally, features should be distinct for different symbol images so that the computer can extract the correct model from the library without confusion. Meanwhile, also it is wanted the features to be robust enough so that they will not be affected by viewing transformations, noises, resolution variations and other factors.

## 1.2 Historical Review

Historically, the Automatic Character Recognition (ACR) systems have evolved in three ages. ***The early age expanded through the interval 1900-1980:*** The ACR studies were started by the Russian scientist Tyurin in 1900 **[Gov90].** The first modern character recognizers appeared in the middle of the 1940s with the development of the digital computers. In this time early work on the automatic recognition of characters has been concentrated either upon printed text or upon small set of handwritten text or symbols. Successful but constrained algorithms had been implemented mostly for Latin characters and numerals. Besides, some studies are published on Japanese, Chinese, Hebrew, Indian, Cyrillic, Greek and Arabic (characters and numerals) in both printed and handwritten cases are also done **[Adi91].** In 1950 the commercial character recognizers appeared with introducing electronic tablets capturing the x-y coordinate data of pen-tip movement this innovation enabled the researchers to work on the on-line handwritten recognition **[Sue80].**

***The middle age was during the decade 1980-1990:*** The studies until 1980 suffered from the lack of advanced Algorithms, powerful hardware and optical devices. With the explosion on the computer hardware and software technology, the previously developed methodologies found a very fertile environment for rapid growth in many application areas as well as ACR system development **[Boz89].**

***The last age begin after 1990 Advancements:*** There is a renewed interest in the character recognition field was found, which involves the recognition of both printed and handwritten characters. Nowadays, it is not only we now have more compact and powerful computers and more accurate electronic equipment such as scanner, camera, electronic tablet etc., but, it was have better recognition algorithms which utilize advanced methodologies such as neural networks, hidden markov models or fuzzy set reasoning. But, still there is a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading.

Since the past few decades a number of researchers have investigated the problem of handwritten digit (character) recognition and many methods have been developed. However, no system to date has achieved the goal of system acceptability. Researchers in this field have proposed different approaches, such as statistical, structural, and moments based approaches **[Cas95].**

In 1987 N. Ula **[Ula87]** used a table look-up for the recognition of isolated handwritten Arabic characters. In this approach, the character is placed in a frame, which is divided into six rectangles and a contour tracing algorithm is used for coding the contour as a set of directional vectors by using a Freeman code. However, this information is not sufficient to determine Arabic characters. Therefore, extra information related to the number of the secondary parts and their position was added. If there is no match, the system will add the feature vector to the table and consider that character as a new entry).

In 1990 S. Dabi **[Dab90]** adopted a statistical approach for recognizing Arabic typewritten characters. In this approach the characters are segmented and recognized by using accumulative invariant moments as feature descriptors.

In 1992 V. Margner **[Mar92]** used outer contours to segment an Arabic word into characters. The word is divided into a series of the curves by determining the start and end points of the word. Whenever the outer contour changes sign a character is segmented.

In 1998 E. J. Mohammed **[Moh95]** design and implement an arabic character recognizer. The complex moment concept is adopted as a method of matching. Different measurement of character feature have been used for identifying the characters, such as Root Mean Square (RMS) distance or gradient parameters.

In 2001 T. O. Robert **[Rob01]** designed a structural recognition approach for extracting morphological features and performing classification without relying on domain knowledge. This system employs a statistical classification technique to perform discrimination based on structural features is a natural solution. A set of morphological features is suggested as the foundation for the development of a suite of structure detectors to perform generalized feature extraction for structural pattern recognition in time-series data.

In 2002 C. Kam-Fai **[Kam02]** proposed a syntactic (structural) approach for the analysis of on-line handwritten mathematical expressions. The authors used Definite Clause Grammar (DCG) to define a set of replacement rules for parsing mathematical expressions.

## 1.3 Aim of the Project

The present work aims at utilize moments invariant descriptors to recognize the isolated Arabic characters. Almost, this is carried out by testing the adopted system on the printed characters as a primary stage, determines the efficiency of the recognition system, and then applies the adopted system on the handwritten Arabic characters.

## 1.4 Thesis Outlines

In addition to current chapter, the remaining chapters in present thesis drop through three chapters. In the following, a detailed explanation on each one of them is found.

♦ **Chapter Two** ***"Character Recognition Approaches"***

This chapter presents the basic concepts of the recognition and discuss the types of approaches.

♦ **Chapter Three** ***"Character Recognition System"***

This chapter deals with the mathematical and programming details of the adopted approach. The used method are explained the results.

♦ **Chapter Four** ***"Conclusion and Suggestion for future work"***

This chapter contains some of conclusions appeared during the time of the current work, which actually result from the analysis of the input/output information. Moreover, it contains some suggestions that should be noted in the future work.

# Chapter Two

# Character Recognition Approaches

# Chapter Two
# Character Recognition Approaches

## 2.1 Introduction

Pattern recognition techniques are among the most important tools used in the field of machine intelligence. The pattern recognition can be defined as the categorization of input data into identifiable classes by extracting significant features or attributes of the data from a background of irrelevant detail **[Gon78].** Thus, the study of pattern recognition problems may logically divide in to two major categories;

1- The study of the pattern recognition capability of human beings and other living organisms.
2- The development of theory and practical techniques for machine implementation of a given recognition task **[Mer05].**

There are two fundamental approaches to implement a pattern recognition system; *statistical* and *structural*. Each approach employs different techniques within the description and classification tasks which constitute a pattern recognition system.

Statistical pattern recognition draws from established concepts in statistical decision theory to discriminate among data from different groups based upon quantitative features of the data. The quantitative nature of statistical pattern recognition makes it difficult to discriminate among groups based on the morphological (i.e., shape based or structural) sub patterns and their interrelationships embedded within the data. This limitation provided the impetus for the development of a structural approach to pattern recognition.

Structural pattern recognition sometimes referred to as syntactic pattern recognition due to its origins in formal language theory, relies on syntactic grammars to discriminate among data from different groups based upon the morphological interrelationships (or interconnections) present within the data.

Structural pattern recognition systems have proven to be effective for data which contain an inherent, identifiable organization such as character or digit recognition **[Taa05].** Handwritten character recognition proves to be challenging problem due to the large variance the data may exhibit, not only there are changes and distortions from one writer to another, but even for samples produced by the same writer . Figure (2.1) shows the different schemes exist under the term character recognition.

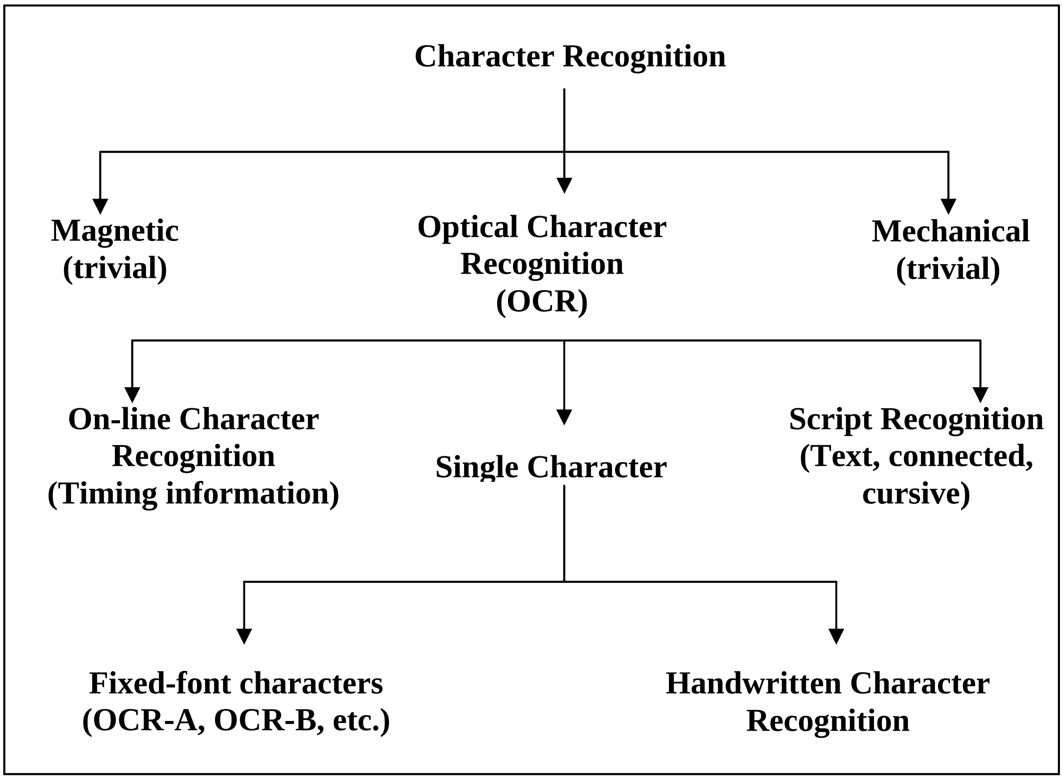



Figure (2.1) General areas covered under term "Character Recognition" **[Moh95]**.

The order of difficulty of character recognition problems is (starting from the most difficult); script recognition, handwritten character recognition, on-line character recognition, fixed-font character recognition and the trivial task of magnetic or mechanical characters.

## 2.2 Machine Learning [Tho02]

Machine Learning is the study of methods for programming computers to learn. Computers are applied to a wide range of tasks, and for most of these it is relatively easy for programmers to design and implement the necessary software. However, there are many tasks for which this is difficult or impossible. These can be divided into four general categories. First, there are problems for which there exist no human experts. For example, in modern automated manufacturing facilities, there is a need to predict machine failures before they occur by analyzing sensor readings.

A machine learning system can study recorded data and subsequent machine failures and learn prediction rules. Second, there are problems where human experts exist, but where they are unable to explain their expertise. This is the case in many perceptual tasks, such as speech recognition, hand-writing recognition, and natural language understanding. Virtually all humans' exhibit expert-level abilities on these tasks, but none of them can describe the detailed steps that they follow as they perform them. Fortunately, humans can provide machines with examples of the inputs and correct outputs for these tasks, so machine learning algorithms can learn to map the inputs to the outputs. Third, there are problems where phenomena are changing rapidly. In finance, for example, people would like to predict the future behavior of the stock market, of consumer purchases, or of exchange rates. These behaviors change frequently, so that even if a programmer could construct a good predictive computer program, it would need to be rewritten frequently. A learning program can relieve the programmer of this burden by constantly modifying and tuning a set of learned prediction rules. Fourth, there are applications that need to be customized for each computer user separately. Consider, for example, a program to filter unwanted electronic mail messages. Different users will need different filters. It is unreasonable to expect each user to program his or her own rules, and it is infeasible to provide every user with a software engineer to keep the rules up-to-date. A machine learning system can learn which mail messages the user rejects and maintain the filtering rules automatically.

Machine learning addresses many of the same research questions as the fields of statistics, data mining, and psychology, but with differences of emphasis. Statistics focuses on understanding the phenomena that have generated the data, often with the goal of testing different hypotheses about those phenomena. Data mining seeks to find patterns in the data that are understandable by people. Psychological studies of human learning aspire to understand the mechanisms underlying the various learning behaviors exhibited by people (concept learning, skill acquisition, strategy change, etc.).

## 2.3. Characteristic of Arabic Alphabets [Tim01]

Due to the cursive nature of the script, there are several characteristics that make recognition of Arabic distinct from the recognition of Latin scripts or Chinese see figure (2.2).

أ ب ت ث ج ح خ

د ذ ر ز س ش ص

ض ط ظ ع غ ف ق

ك ل م ن

ه و ي

Figure (2.2) Letters of the isolated Arabic alphabet.

Arabic has 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.

Arabic is a cursive language. There are no capital letters and some letters are not connected to the letters that follow them. Thus, words cannot be segmented based on pen-up/pen-down information or space between letters. Block or hand printed letters do not exist in Arabic. In summary, many researchers have been working on cursive script recognition for more than three decades. Nevertheless, the field remains one of the most challenging problems in pattern recognition and all the existing systems are still limited to restricted applications. Arabic has four forms for each letter depending on the position of the letter in each word. These are initial, medial, final and isolated. A key difference between Latin scripts and Arabic is the fact that many letters only differ by a dot(s) but the primary stroke is exactly the same. This highlights the need for a good feature extractor/classifier for the secondary stroke(s).

## 2.4 Automatic Character Recognition Systems

In this section, an overview of Automatic Character Recognition (ACR) techniques is presented; Data is captured from its source to the computers by various data acquisition techniques. The ACR methodologies are very much dependent on the type of the equipment used for data acquisition. The text type is another determining factor for the ACR techniques. In the following, the classification of the ACR techniques will be given.

## 2.4.1 Systems Classified According to Data Acquisition

The progress in Automatic Character Recognition systems is evolved in two categories according to the mode of data acquisition:

- On-line character recognition systems.
- Off-line character recognition systems.

Off-line character recognition captures the data from paper through optical scanners or cameras whereas the on-line recognition systems utilize the digitizers which directly capture writing with the order of the strokes, speed, pen-up and pen-down information. **[Naf98].**

### 2.4.1.1 On-line Character Recognition Systems [Tia91]

The problem of recognizing handwriting recorded with a digitizer as a time sequence of pen coordinates is known as on-line character recognition. Whenever the digitizer captures the data during writing, the ACR system with or without a lag makes the recognition. The digitizers are mostly electromagnetic-electrostatic tablets which send the coordinates of the pen tip to the host computer at regular intervals. Some digitizers use pressure-sensitive tablets which have layers of conductive and resistive material with a mechanical spacing between the layers. There are also, other technologies including laser beams and optical sensing of a light pen.

While nothing opposes the idea of a computer that would use multiple input modalities including speech, keyword and pen, some applications call for a pen-only computer interface; in social environment, speech does not provide enough privacy, for small hand-held devices and for large alphabet, the keyboard is cumbersome. Applications are numerous; personal organizer, personal communicator, notebook, data acquisition device for order entries, inspections, inventories, surveys, etc.

The on-line handwriting recognition problem has a number of distinguishing features, which must be exploited to get more accurate results than the off-line recognition problem;

- It is adaptive. The immediate feed-back is given by the writer whose corrections can be used to further train the recognizer.
- It is a real time process. It captures the temporal or dynamic information of the writing.

On the other hand, the disadvantages of the on-line character recognition are as follows

- The writer requires special equipment, not as pen and paper which is comfortable and natural to use.
- It can not be applied to documents printed or written on papers.
- Punching is much faster and easier than handwriting for small size alphabet such as English or Arabic.

### 2.4.1.2 Off-line Character Recognition Systems [Naf98]

Off-line character recognition is also, known as "Optical Character Recognition" (OCR); because the image is converted into a bit pattern by an optically digitizing device such as optical scanner or camera. The recognition is done on this bit pattern data for both printed and hand-written text. The research and development is well progressed for the OCR of the printed documents.

The bit pattern data is shown by a matrix of pixels. This matrix can be very large. In order to meet the complexity and sophistication and to insert much data in recognition, most scanners are designed to have an x-y resolution of typically 100-1600 dots per inch.

The major advantage of the off-line recognizers is to allow the previously written and printed texts to be processed and recognized. Some applications of the off-line recognition are large scale data processing such as postal address reading, cheque sorting, short hand transcription, automatic inspection and identification, reading aid for visually handicapped.

The drawbacks of the off-line recognizers, compared to on-line recognizers are summarized as follows:

- Off-line conversion usually requires costly and imperfect pre-processing techniques prior to feature extraction and recognition stages.
- They do not carry temporal or dynamic information such as the number and order of pen-on and pen-off movements, the direction and speed of writing and in some cases, the pressure applied while writing a character.
- They are not real-time recognizers.

### 2.4.2 Systems Classified According to Text Type [Tia91]

There are two main areas of interest in character recognition, namely:

- Printed Character Recognition.
- Hand-written Character Recognition.

The problem of printed character recognition is relatively well understood and solved with little constraints. When the documents are typed on a high quality paper with modern printing technologies. On the other hand, hand-written character recognition systems have still limited capabilities even for recognition of the Latin characters.

#### 2.4.2.1 Printed Character Recognition

The printed texts include all the printed materials such as books, newspapers, magazines, and documents which are the outputs of typewriters, printers or plotters. On the basis of the capabilities and complexities printed characters can be further classified as:

- Fixed-font Character Recognition (recognition of a specific font).
- Multi-font Character Recognition (recognition of more than one font).
- Omni-font Character Recognition (recognition of any font).

The above classes of printed character recognition problems are well studied. However, the recognition rates of the commercially available products are very much dependent on the age of the documents, quality of paper and ink which may result in significant data acquisition noise.

#### 2.4.2.2 Hand-Written Character Recognition

Handwritten character recognition, based on the form of written communication, can be divided into two categories- cursive script and hand-printed characters. In practice, however, it is difficult to draw a clear distinction between them. A combination of these two forms can be seen frequently. Based on the nature of writing and the difficulty of segmentation process.

## 2-5 Character Recognition Paradigm

Since there are common aspects between group of some techniques, and other common aspects between other group of techniques. Therefore these technique can be classified into three major approaches based upon the common aspects of groups, these are:

### 2.5.1 Statistical Approach

In the statistical approach, each pattern is represented in terms of (*n*) features or measurements and is viewed as a point in a d-dimensional space. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d-dimensional feature space. The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated. Given a set of training patterns from each class, the objective is to establish decision boundaries in the feature space which separate patterns belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions of the patterns belonging to each class, which must either be specified or learned **[Dev96].**

### 2.5.2 Geometrical Approach (Test Method)

The most common aspects of all techniques used in this approach are geometrical features such as, line, angles of intersected lines, curves, etc. Therefore series of tests are applied in order to determine distinctive features in the unknown character which are compared to that learned characters. This approach is specific for a given application whereby the tests are designed for each particular problem. In the case where the number of classes to be identified is great, this approach becomes difficult to apply **[Pav75].**

### 2.5.3 Template Matching Approach

One of the simplest and earliest approaches to pattern recognition is based on template matching. Matching is a generic operation in pattern recognition which is used to determine the similarity between two entities (points, curves, or shapes) of the same type. In template matching, a template (typically, a 2D shape) or a prototype of the pattern to be recognized is available. The pattern to be recognized is matched against the stored template while taking into account all allowable pose (translation and rotation) and scale changes. The similarity measure, often a correlation, may be optimized based on the available training set. Often, the template itself is learned from the training set. Template matching is computationally demanding, but the availability of faster processors has now made this approach more feasible **[Rob00].**

### 2.5.4 Syntactical (Structural) Approach

The recognition in this approach is done through a description of the pattern in terms of primitive elements and relations between them. *The syntactical technique* makes use of the results obtained from the theory of formal languages which have been developed in other areas of computer science. The important aspect of this method is the possibility of using a grammar and its recursive nature to describe a pattern by a string of characters. The first applications of this method are the coding of contours in the images (chromosomes…) this approach has been generalized to three-grammar **[Fu82]**. Meanwhile *the structural technique* is similar to the syntactical technique in that it makes use of a graph representation of a pattern; the main difference is, however, the absence of a grammar. The necessity of such representation and has been recognized since the early structural techniques **[Rot76].**

### 2.5.5 Moments Based Approach

The seven Moments constitute an important set of parameters for image analysis. The low order moment contain significant information about simple object. They have been used in finding the location and orientation of an object. Moment invariants have been used as features for pattern recognition. The $0^{th}$ order moment represent the total mass (area for binary image) of the object. The first order moments locate the centroid of an object in the image. The angle of the principle axis, which characterizes the orientation of the object in the image, can be found from the lower order moments up to the second-order moments. The location, size and orientation are the fundamental description of an object. They have the desirable property of being invariant under the object translation, scaling and rotation **[Kep90].** However, since moments have such power of describing an object, which requires certain mathematical notations, it was selected to contribute in building the proposed character recognition system as presented in chapter three.

## 2.6 Proposed Character Recognition Systems

The designed character recognition system based on the series of operations needed to construct the optical character recognition, the functional structure of pattern recognition system show in figure (2.3) the following sections will show more details about these operations.

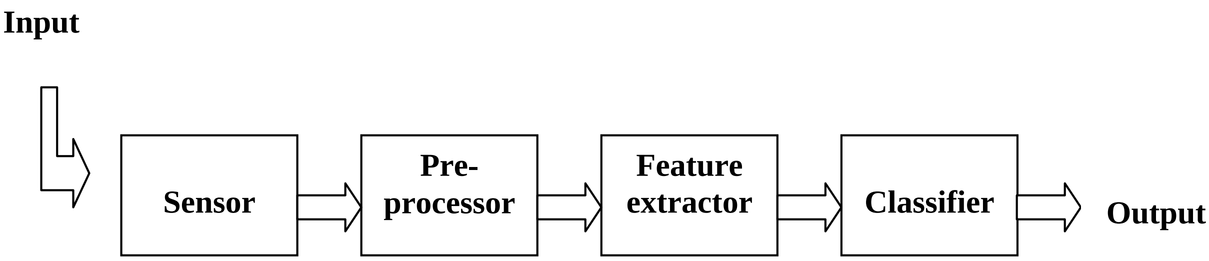



Figure (2.3) Functional structure of pattern recognition system **[Lei07]**.

### 2.6.1 Sensor **[Lei07]**

The conversion from printed page to digital image often involves specialized hardware like an optical scanner that attempts to determine the color value at evenly spaced points on the page. The scanning resolution will determine how many of these points will be inspected per unit of page length.

Typically this is specified in dots or pixels per inch, thus a document scanned at a resolution of 300 Dot Per Inch (DPI) will have been sampled at 300 evenly spaced points for each inch of each page. At each of these sample points or pixels the color depth will determine to what extent the recognized color matches the true color on the page. One of the more common color depths involves using a single byte to store each of the red, green and blue channels (plus an optional additional byte used to store opacity information). Since each byte of storage is made up of 8 bits, $2^8 = 256$ unique shades of a color can be represented. Since any color can be created by mixing red, green, and blue, the desired color is approximated using the closest shade of red, green, and blue. Note that color depth is often measured in Bits Per Pixel (BPP).

Image that contains the text to be recognized is acquired by using scanner and a bitmap image with true colors will be saved for further processing.

## 2.6.2 Pre-processing

The pre-processing stage includes the following pre-processes:

1- Noise Removal.

2- Binarization.

3- Segmentation.

### 2.6.2.1 Noise Removal

During the scanning process, differences between the digital image and the original input (beyond those due to quantization when stored on computer) can occur. Hardware or software defects, dust particles on the scanning surface, improper scanner use, etc. can change pixel values from those that were expected. Such unwanted marks and differing pixel values constitute noise that can potentially skew character recognition accuracy.

Furthermore, certain marks or anomalies present in the original document before being scanned (from staples, or telephone line noise during fax transmission etc.) constitute unwanted blemishes or missing information that one may also like to rectify and correct before attempting character recognition. In this project the *median filtering* is used **[Lei07].**

In *median filtering* the input pixel is replaced by the median of the pixels contained in the neighborhood, the algorithm for median filtering requires arranging the pixel gray values in the neighborhood in ascending or descending order and picking up the value at the center of the array. Generally the size of the neighborhood is chosen as odd number so that a well-defined center value exists. If, however, the size of the neighborhood is even the median is taken as the arithmetic mean of the two values at the center. Some of the properties of median filter are:

- It is a nonlinear filter.
- It is useful in removing isolated lines or pixels while preserving spatial resolution. It is found that median filter works well on binary noise but not so well when the noise is Gaussian.
- Its performance is poor when the number of noise pixels is greater than or equal to half the number of pixels in the neighborhood **[Hua79].**

### 2.6.2.2 Image Binarization

Images in this stage are considered to be binary. The pixels in binary image can assume only two values, 0 or 1; the binary images are thus least expensive, since the storage and also processing requirement is least. Examples of binary images are line drawings, printed text on a white page, or silhouette. These images contain enough information about the objects in the image that can recognized easily. There are a number of applications in computer vision where binary images are used for object recognition, tracking, and so on. The applicability of binary images is, limited because the overall information content in such images is limited **[Jai95].**

A gray level image may be converted to a binary image by thresholding process. The process of converting the image into binary mode is called binarization.

Several approaches to binarization have been discussed in the literature but they typically fall into one of two categories. Global methods treat each pixel independently, converting each to black or white based on a single threshold value. If a pixel’s color intensity is higher than the global threshold it is assigned one value, otherwise it is assigned the opposite value. In contrast local methods, make use of the color information in nearby pixels to determine an appropriate threshold for a particular pixel. The goal of the binarization is to separate the character from the background in the gray image and reduce the image color into Black & White. **[Sna04].**

#### 2.6.2.3 Image segmentation

It is generally the first step in any attempt to analyze or interpret an image automatically. It is defined to be the process of partitioning the image into distinct region that are meant to correlate strongly with objects or features of interests in the image. It also regarded as a process of grouping together pixels that have similar attributes.

### 2.6.3 Feature Extraction [Mar05]

Feature is a set of information represent the observed data efficiently such that minimizing the within class pattern variability while enhancing the between class pattern variability.

There are two main types of features:

1- Statistical (Numerical, Quantitative) features

Input data → Feature Extraction → Feature Vector

2- Structural (Syntactical, Quantitative) features

Input data → Feature Extraction → Primitive Structure

Feature extraction process will generate feature sets that represent characters occurrences in the image. In current work, features are extracted by using ***character moments***. In order to recognize an object in an image, we need to make two choices. The first one concerns the selection of a characteristic feature of each object. This feature must have in general some properties as invariance by rotation, scale or translation of the object, which is appropriate to the objective of the current thesis.

It can be directly computed on the original image or after wards segmentation result as for example a contour detection. The second choice concerns the decision criteria for the object recognition among one of its known objects in the knowledge database using the previous features.

### 2.6.3.1 Shape Features [Tin05]

Shape is another image feature. Shape can roughly be defined as the description of an object minus its position, orientation and size. Therefore, shape features should be invariant to translation, rotation, and scale, when the arrangements of the objects in the image are not known in advance. To use shape as an image feature, it is essential to segment the image to detect object or region boundaries; and this is a challenge. Techniques for shape characterization can be divided into two categories. The first category is ***boundary-based***, using the outer contour of the shape of an object. The second category is ***region-based***, using the whole shape region of the object. The most prominent representatives of these two categories are Fourier descriptors and moment invariants. The main idea behind the Fourier descriptors is to use the Fourier-transformed boundaries of the objects as the shape features, whereas the idea behind moment invariants is to use region-based geometric moments that are invariant to translation and rotation.

### 2.6.3.2 Hu's Moment Computation [Qin03]

In region-based invariants, all of the pixels of the image are taken into account to represent the shape. Because region-based invariants combine information of an entire image region rather than exploiting information just at the boundary points, they can capture more information regarding the image.

Moment-based invariants are the most common region-based image invariants which have been used as pattern features in many applications. Hu in 1961 introduced a set of moment-based invariants using nonlinear combinations of regular moments. Hu's seven moments invariants have the desirable properties of being invariant under image translation, scaling, and rotation.

Hu identified seven normalized central moments as shape features, which are also scale invariant. These seven invariants are expressed by the following equations:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\ dxdy \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.1)$$

*For p,q=0,1,2…………*

Where $m_{pq}$ is the $(p+q)^{th}$ order moment of the continuous image function $f(x,y)$.

The central moments of $f(x,y)$ are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)dxdy \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.2)$$

Where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$, which are the centroid of the image.

For digital images the integrals are replaced by summations and $m_{pq}$ becomes:

$$\mu_{pq} = \sum_x \sum_y x^p y^q f(x,y) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

Then the central moments are changed to:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.4)$$

The central moments of order 3 are as follows:

$$\mu_{10} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^0 f(x,y) = m_{10} - \frac{m_{10}}{m_{00}} m_{00} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.5)$$

$$\mu_{11} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^1 f(x,y) = m_{11} - \frac{m_{10}}{m_{00}} m_{01} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

$$\mu_{20}=\sum_x\sum_y(x-\bar{x})^2(y-\bar{y})^0 f(x,y) = m_{20}-\frac{2m_{10}^2}{m_{00}}+\frac{m_{10}^2}{m_{00}} = m_{20}-\frac{m_{10}^2}{m_{00}} \quad \text{.......(2.7)}$$

$$\mu_{02}=\sum_x\sum_y(x-\bar{x})^0(y-\bar{y})^2 f(x,y) = m_{02}-\frac{m_{01}^2}{m_{00}} \quad \text{.........................(2.8)}$$

$$\mu_{30}=\sum_x\sum_y(x-\bar{x})^3(y-\bar{y})^0 f(x,y) = m_{30}-3\bar{x}m_{20}+2m_{10}\bar{x}^2 \quad \text{............(2.9)}$$

$$\mu_{12}=\sum_x\sum_y(x-\bar{x})^1(y-\bar{y})^2 f(x,y) = m_{12}-2\bar{y}m_{11}-\bar{x}m_{02}+2\bar{y}^2 m_{10} \quad \text{.......(2.10)}$$

$$\mu_{21}=\sum_x\sum_y(x-\bar{x})^2(y-\bar{y})^1 f(x,y) = m_{21}-2\bar{x}m_{11}-\bar{y}m_{20}+2\bar{x}^2 m_{01} \quad \text{.......(2.11)}$$

$$\mu_{03}=\sum_x\sum_y(x-\bar{x})^0(y-\bar{y})^3 f(x,y) = m_{03}-3\bar{y}m_{02}+2\bar{y}^2 m_{01} \quad \text{...........(2.12)}$$

The normalized central moments, denoted by $\eta_{pq}$, are defined as:

$$\eta_{pq}=\frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{................................................(2.13)}$$

Where $\gamma=(p+q)/2+1$ ................................... (2.14)

For $p+q=2,3....$ see figure (2.4).

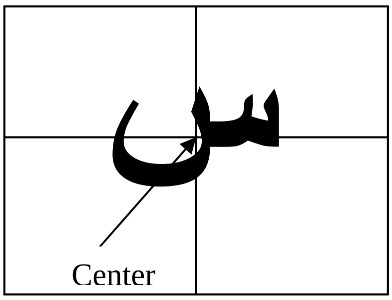



Figure (2.4) Normalized image of character (س)

From the second and third moments, a set of seven invariant moments can be derived. They are given by:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.15}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.16}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.17}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.18}$$

$$\left.\begin{aligned}\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned}\right\} \tag{3.19}$$

$$\left.\begin{aligned}\phi_6 &= (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\ &+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned}\right\} \tag{2.20}$$

$$\left.\begin{aligned}\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ &+ (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned}\right\} \tag{2.21}$$

Figure (2.5) shows the behavior for Hu’s seven Moments values for Arabic character (س).

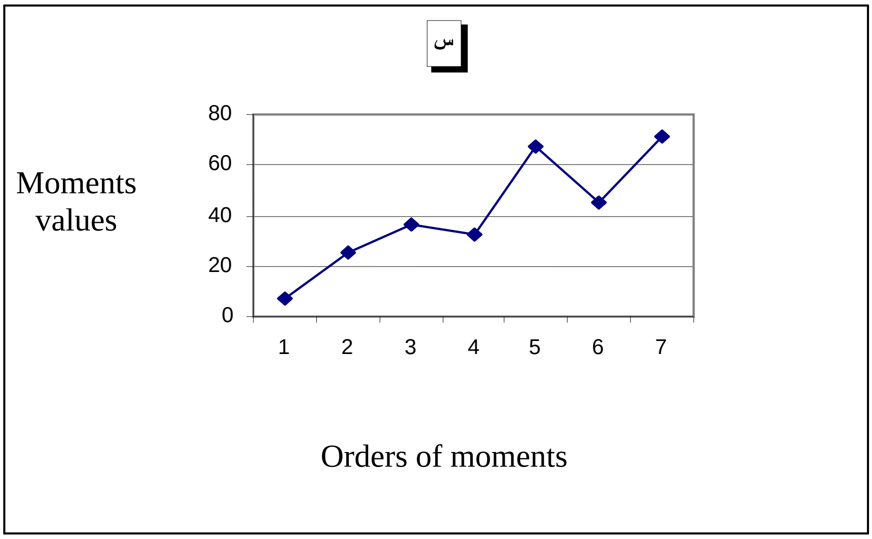



Figure (2.5) Hu's seven Moments Behavior of Arabic Character (س)

## 2.7 Comparable Model

When working with image representations of symbols from an alphabet, the similarity should attempt to classify each symbol to a specific class.

Since each symbol image is represented by a matrix of pixel intensity values located by the co-ordinates of its rectangular bounding box, similarity should be calculated using this information in some manner. Often, this is carried out by converting the $m$ row by $n$ column matrix representation into a single vector of pixels of length $m \times n$. These vectors can then be thought of as defining points in an $m \times n$ dimensional Euclidean space, where each pixel's value contributes to the final point location along a single dimension. By representing each symbol image as a point in Euclidean space, this allows us to determine similarity of two images based on the distance between their defined points.

In order to recognize an object in an image, it was need to make two choices. The ***first*** one concerns the selection of a characteristic feature of each object. This feature must have in general some properties as invariance by rotation, scale or translation of the object.

It can be directly computed on the original image or afterwards segmentation result as for example a contour detection. The ***second*** choice concerns the decision criteria for the object recognition among one of its known objects in the knowledge database using the previous features.

The Euclidean distance metric is typically what one person appeals to when the term distance is used in its natural or everyday sense. The Euclidean distance between two points is measured as the length of a straight line used to connect them. For two symbol image vectors A and B both composed of n pixels, this distance DE is formally defined in Equation (2.22) **[Lei07].**

$$DE = \sqrt{\sum_{i=1}^{n} (A(i) - B(i))^2} \quad \text{.....................................} \ (2.22)$$

The quantities *A(i) and B(i)* in Equation (2.22) denote the intensity value of the $i^{th}$ pixel in image *A* and *B* respectively.

# Chapter Three

# Character Recognition System

# Chapter Three
# Character Recognition System

## 3.1 Introduction

The principal function of a pattern recognition system is decision-making concerning the class membership of the pattern, and with which class to be compared. Pattern recognition system involves several major functions starting from the input pattern until the deciding which class the pattern belongs to.

The sensor function is simply the measurement device that transforms the input pattern into a form suitable for machine manipulation. Although some simple pattern recognition methods operate on the input data directly from the sensor, it is common practice to follow the sensor with a preprocessor and feature extractor. Moreover, the preprocessing function removes unnecessary or corrupting elements from the measured data, while the feature extractor computes from preprocessed data, for the features required in classification. Finally, these features are input into the classifier, whose function is to yield a decision concerning the class membership of the pattern being processed. This chapter describes the method that adopted in the character recognition process, which depends on the theoretical principles and mathematical expressions mentioned in previous chapter. The most popular Arabic font are Thuluth, Koufi, Naskh, Dewani and Rokaa. Each of them has its typeface that can be recognized from others. Rokaa font is near to the traditional font of arabic characters, which is used in this thesis. Figure (3.1) shows the stages needed to constraint the character recognition .

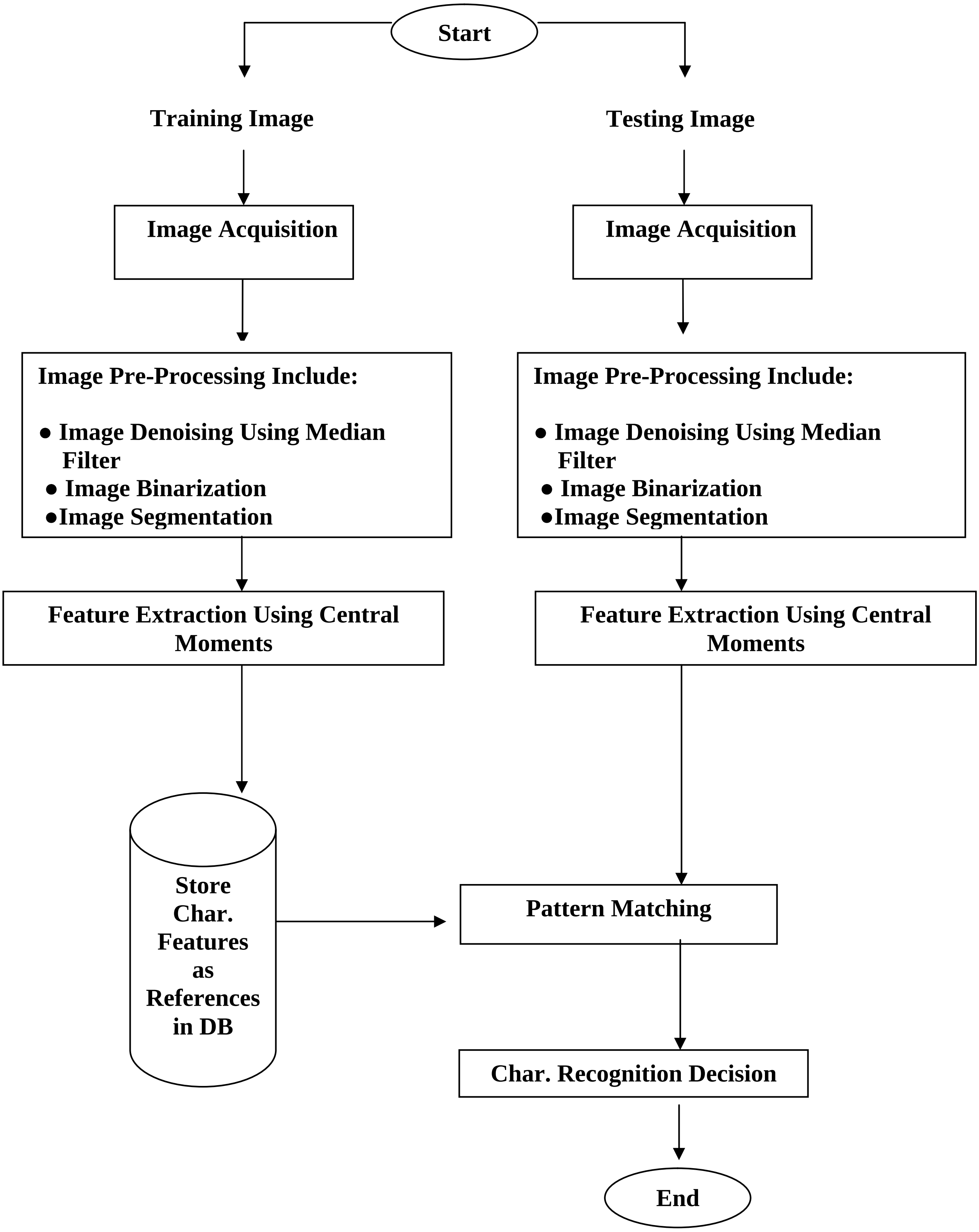



Figure (3.1) Series of operations needed to construct the optical character recognition.

The adopted method passes through some of image preparing processes. The following steps point out the main sequenced operations of the present work.

1- Get an image have multi-characters within.
2- Separate each character by determining a minimum box that encloses the character.
3- Normalized each sub-image by determining its size.
4- Extract the features from each sub-image.
5- Match the extracted features with those existing in the library to determine the type of character.
6- Repeat steps 4 and 5 for each sub-image until end of them.

## 3.2 Image Acquisition

In image acquisition process the images of characters are scanned using scanner device of type (***canon***) and using white papers of type (***A4***), the image are scanned in ***300*** dpi resolutions and saved as 8 bit per pixel bitmap.

## 3.3 Image Denoising (Algorithm and Results)

The denoising process will remove the noise coming during scanning process. Three filter types are adopted in this work Median filter, Minimum filter and Maximum filter with mask filter size ($3\times3$) to choose the value of each pixel from the values around it.

It was found that using median filter for the images that contains characters gives good results than using the maximum filter or minimum filter, because when the minimum filter was used noise will be increased and characters in the image will be like a bold characters and this case will cause problems in the segmentation process, also when the maximum filter was used the characters in the image will loss some parts of them, and this will be also cause another problems in the segmentation process, so that the median filter was used to remove the noise from image of characters.

Algorithm (3.1) show how to perform the median filtering process on the image, this algorithm performs the following:

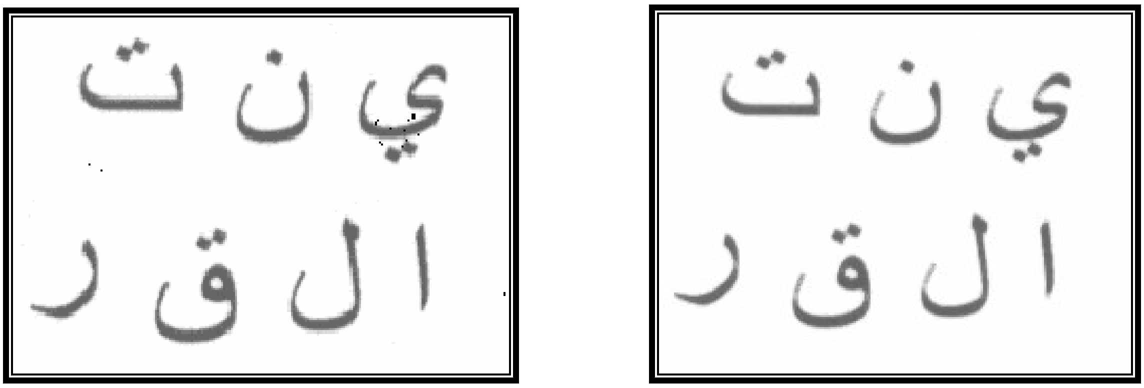
1- Perform a mask of size (3×3) on the image data, transfer the values to a one dimensional array.
2- Sort the values of array ascendingly or descendingly.
3- Choose the median value from the array as new value of output image.

Figures (3.2) and (3.3) show the images before and after denoising process for both printed and hand written Arabic characters respectively.

**<u>Algorithm (3.1) Denoising Filter</u>**

**Input:**
P_DATA (): image data
H : height of the image
W : width of the image
**Output:** image with out noise (Change the contents of P_DATA())

**1- For** all i, j **Do** {where 1≤i≤H-2 and 1≤j≤W-2}
**2- Set z← 0**
**3- For** all x, y **Do**{where 0≤ x≤2 and 0≤y≤2}
**4- Set** Median(z) ← P_DATA(x + i - 1, y + j - 1)
**5- Set z ← z + 1**
**6- End for**
**7- For** all x, y **Do**{where 0≤ x≤ z-2 and x+1≤y≤z-1} **//Sort Mask**
**8-** Check **If** Median(x) > Median(y) **Then**
**9-** Median(x) ↔ Median(y)// swapping process
**10- End If**
**11- End for**
**12- Set** Buffer(i, j) ← Median (z / 2)
**13-End for**
**14-For** all i, j **Do** {where 0≤i≤H-1 and 0≤j≤W-1}
**15- Set** P_DATA(i, j) ← Buffer(i, j)
**16-End for**
**17-End.**

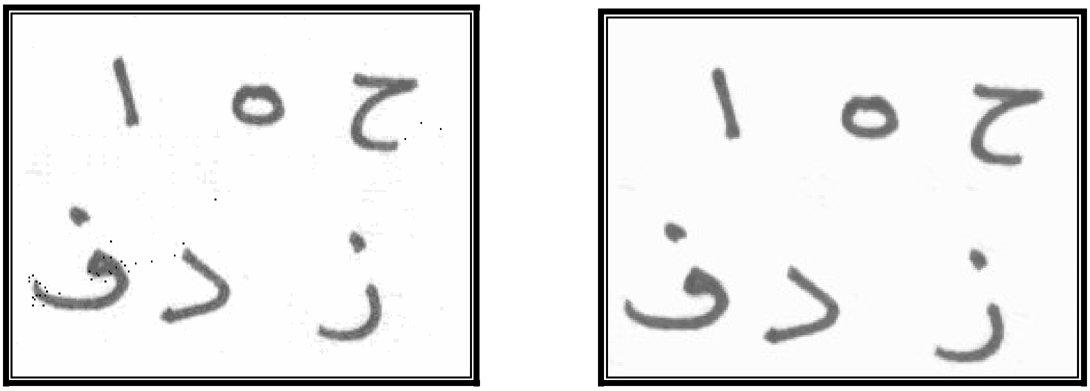


(a) (b)

Figure (3.2) Image of printed Arabic characters.

(a) Before denoising process.

(b) After denoising process.



(a) (b)

Figure (3.3) Image of hand written Arabic characters.

(a) Before denoising process.

(b) After denoising process.

## 3.4 Image Binarization (Algorithm and Results)

In the binarization process the image will be converted to binary image (i.e. black and white) using the adaptive threshold, it was impotent perform the binarization process after denoising process.

If the binarization process doing directly from the scanner device the noise will still on the paper, so that the binarization process performed after denoising process.

Algorithm (3.2) shows how to perform the binarization process, this algorithm can be performs as follows:

1- Find the average (*T*), of the image.
2- Estimate the summation ($M_1$) of values that greater than *(T)* and count their number ($NuM_1$).
3- Estimate the summation ($M_2$) of values that less than or equal to *(T)* and count their number ($NuM_2$).
4- Compute $Z_1= M_1/ NuM_1$, and $Z_2= M_2/ NuM_2$.
5- The threshold is $T=(Z_1+Z_2)/2$.
6- According to the threshold, each pixel in the image will be transfer to either 0 (black) or 1 (white). Figures (3.4) and (3.5) show the images before and after binarization process for both printed and hand written Arabic characters respectively.


(a) (b)

Figure (3.4) Image of printed Arabic characters.

(a) Before binarization process.

(b) After binarization process.

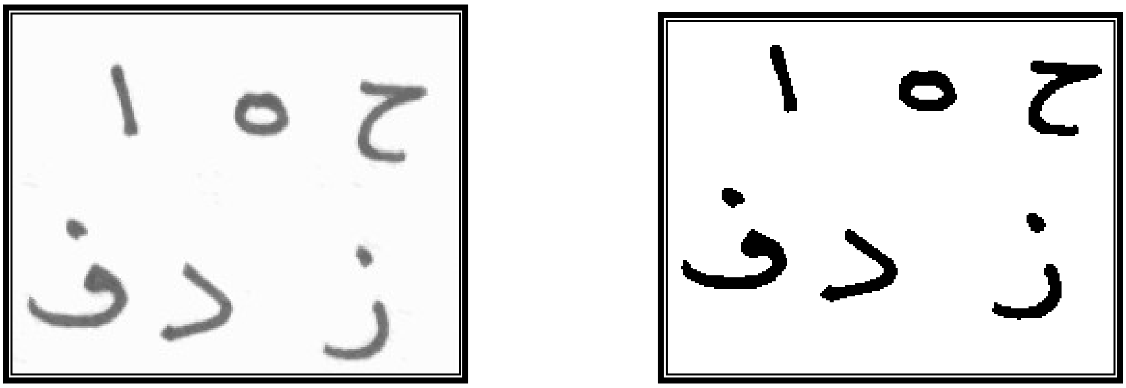
## Algorithm (3.2) Image Binarization

**Input:**

P_DATA () : image data (results from denoising process)
H : height of the image
W : width of the image

**Output:** Binary image Change the contents of P_DATA() to either zero or one)

**1- Set** Average←**128**
**2- set** T ← Average
**3- Do Until Abs(T - T0) < 1**
**4- initialize (** M2, NuM1, NuM2, M1 ) all to zero
**5- For** all i, j **Do** {where 0≤i≤H-1 and 0≤j≤W-1}
**6-** Check **If** P_DATA(i, j) > T **Then**
**7-** **Set** M1 ← M1 + P_DATA(i, j)
**8-** **Increment** NuM1 by 1
**9-** **Else**
**10-** **Set** M2 ←M2 + P_DATA(i, j)
**11-** **Increment** NuM2 by 1
**12-** **End If**
**13- End for**
**14-Set** M1 ← M1 / NuM1**: Set** M2 ← M2 / NuM2
**15- Set** T0 ← T**: Set** T ← 0.5 * (M1 + M2)
**16-** **loop**
**17-For** all i, j **Do** {where 0≤i≤H-1 and 0≤j≤W-1}
**18-** Check **If** P_DATA(i, j) < T **Then**
**19-** **Set** P_DATA(i, j) ← 0
**20-** **Else**
**21-** **Set** P_DATA(i, j) ← 1
**22-** **End If**
**23-End for**
**24-End.**



(a) (b)

Figure (3.5) Image of handwritten Arabic characters.

(a) Before binarization process.

(b) After binarization process.

## 3.5 Image Segmentation (Algorithms and Results)

In the segmentation process each character in the image will be determined by minimum box around it to process each character separately; to perform the process of segmentation must perform the following:

1- Extract image data from the pre-processed image.
2- Determine the upper and lower boundaries for each line of characters in the image by algorithm (3.4).
3- Determine the left and right boundaries for each character in every line obtained from above step by algorithm (3.5).
4- Determine the upper and lower boundaries for each character obtained in each line so that the minimum box can be determined by algorithm (3.6).
5- Create an image for each character by algorithm (3.7).

After enclosed each character by its minimum box normalized the size of each image of characters. While the size of characters increased during the image acquisition, It was found that the size ($32\times32$) is not enough for the segmented characters because the image of characters was scanned with 300 dpi resolution so that the size of characters will be as large as it before the scanning process. And also found that the size ($64 \times 64$) not enough for all characters. In this work the size of window (image) of each character will be normalized to ($128 \times 128$).

It was note that there are characters have dot(s) with it like**:** **(ي،ن،ق،ف،غ،ظ،ض،ش،ز،ذ،خ،ج،ث)** and there are another characters have second part with it like (أ،آ،ؤ,ك) in the segmentation process the dot (or dots) and the second part of the character will not be separate from the body of the character. The number of pixels between dot or second part and the body of the character considered to be up to 7 pixels and this also depends on the font size and font type. In the case of hand written characters the number of pixels between the characters and their dots difficult to be expected where each character can be written in different way and different size. Algorithm (3.3) shows how the segmentation process was performed.

**Algorithm (3.3) Character Segmentation**

**Input:** P_DATA(): image data (results from binarization process)
H : height of the image
W : width of the image

**Output:** Create an image for each character

**1-**Get IMAGE DATA from the Preprocessed Image
**2-** Perform row scan for pixels data equal to zero (i.e., black pixel) to count how many lines of characters in the pre-processed image by algorithm (3.4).

**3-** Perform column scan for the output from the above step to determine the left and right boundaries for each character in every line of image by algorithm (3.5).

**4-** Perform row scan again for each character segmented vertically in the above step by algorithm (3.6).

**5-** Generate an image for each segmented character by algorithm (3.7).
**6- End.**

**Algorithm (3.4) Row scan**

**Input:** P_DATA (): image data (binary image results from binarization process)
H : height of the image
W : width of the image

**Output:** NoofLines // Number of lines in image

```
For all i Do {where 0≤i≤H-1}        //Initialize record of Lines (U: Up , D: Down) to -1
    Set Lines(i).u ← -1: Set Lines(i).D ← -1
End for
Set NoofLines ← 0
For all i, j Do {where 0≤i≤H-1 and 0≤j≤W-1}
    Check If P_DATA(i, j) = 0 Then
        Check If Lines(NoofLines).u = -1 Then
            Set Lines(NoofLines).u ← i
        End If
    Else
        Check If Lines(NoofLines).u <> -1 Then
            Set Lines(NoofLines).D ← i: increment NoofLines by 1
        End If
    End If
End for
Set R ← 0
For all i,j Do {where 0≤ i≤ NoofLines – 2 and i + 1 ≤j≤ NoofLines – 1}
    Check If Lines(i).D = Lines(j).u - 1 Then
        Set Lines(i).D ← Lines(j).D
        Check If j = NoofLines - 1 Then
            Set Lines(R).u ← Lines(i).u: Set Lines(R).D ← Lines(i).D
            Increment R by 1: Set i ← j - 1
        End If
    Else
        Set Lines(R).u = Lines(i).u: Set Lines(R).D ← Lines(i).D
        Increment R by 1: Set i ←j - 1
    End If
End for
Set NoofLines ← R
```

After the number of lines in image was determined in the row scan process, then perform column scan on this image. In this step the left and right boundaries for each character in every line in image will be determined. Figures (3.6) and (3.7) show the images before and after segmentation process for both printed and hand written Arabic characters respectively.

## Algorithm (3.5) column scan

**Input:** NoofLines :Number of lines in image results from algorithm (3.4)
H: height of the image
W: width of the image
**Output:** NoofChar **:** Number of characters

**1-initilize** L**,** NoofChar, TotalChar , Counter **to zero**
**2- Set** CharHW ← 128: **Set** CharHeader ← Header
**3-For** all L **Do**{where 0 ≤L≤ NoofLines – 1}
**4-** **Set** Si ← Lines(L).u: **Set** Ei ← Lines(L).D**:** **Set** NoofChar ← 0
**5-** **For** all i **Do**{where 0 ≤i≤ 1023}
**6-** **Set** CharBorders(i).u ← -1: **Set** CharBorders(i).L ← -1
**7-** **Set** CharBorders(i).D = -1**:** **Set** CharBorders(i).R ← -1
**8-** **End for**
**9-** **For** all j,i **Do** {where 0 ≤j≤ W – 1 and Si ≤i≤To Ei } **//Char Borders**
**10-** Check **if** P_DATA(i, j) = 0 **Then**
**11-** Check **If** CharBorders(NoofChar).L = -1 **Then**
**12-** **Set** CharBorders(NoofChar).L ← j
**13-** **End If**
**14-** **Else**
**15-** Check **If** CharBorders(NoofChar).L <> -1 **Then**
**16-** **Set** CharBorders(NoofChar).R ← j**: Set** CharBorders(NoofChar).u ← Si
**17-** **Set** CharBorders(NoofChar).D ← Ei **: increment** NoofChar by 1
**18-** **End If**
**19-** **End If**
**20-** **End for**
**21-** **Set R ← 0**
**22-** **For** all i,j **Do** {where 0 ≤i≤ NoofChar – 1 and i + 1 ≤j≤ NoofChar**}**
**23-** Check **If** CharBorders(i).R = (CharBorders(j).L - 1) **Then**
**24-** **Set** CharBorders(i).R ← CharBorders(j).R
**25-** Check **If** j = NoofChar **Then**
**26-** **Set** CharBorders(R).L ← CharBorders(i).L
**27-** **Set** CharBorders(R).R ← CharBorders(i).R
**28-** **Increment** R by 1**: Set** i ← j - 1
**29-** **End If**
**30-** **Else**
**31-** **Set** CharBorders(R).L ← CharBorders(i).L
**32-** **Set** CharBorders(R).R ← CharBorders(i).R
**33-** **Increment** R by 1**: Set** i ← j - 1
**34-** **End If**
**35-** **End for**
**36-Set** NoofChar ←R
**37- End For**

**Algorithm (3.6) Next row scan**

**Input:** NoofChar: Number of characters results from the algorithm (3.5)

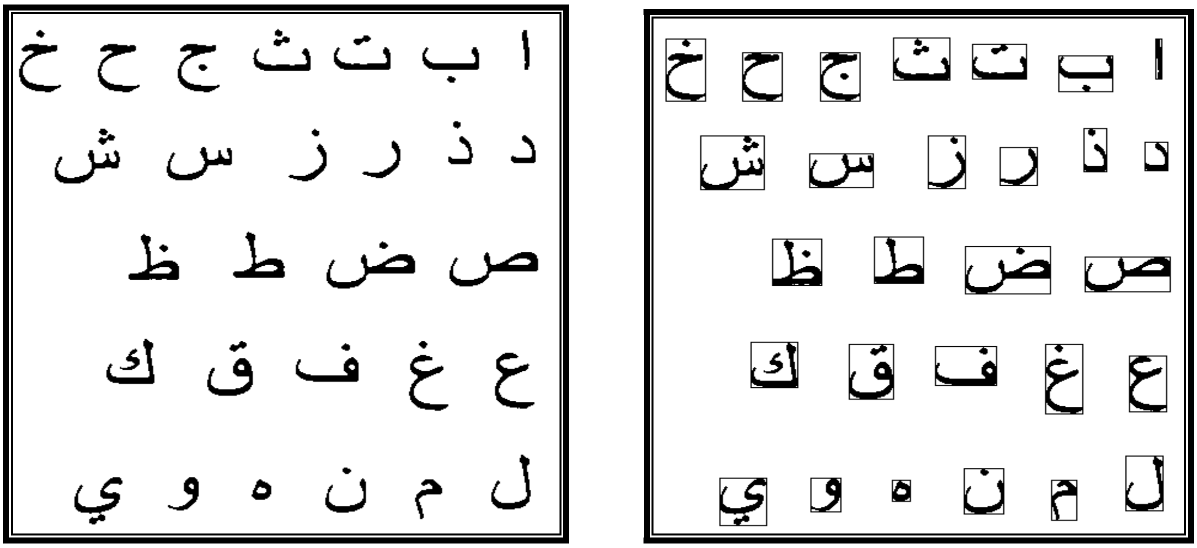
**Out put:** updata the up and down for each character

**1-** **For** all k **Do** {where 0 ≤K≤ NoofChar – 1}
**2-** **Set** Sj ← CharBorders(k).L
**3-** **Set** Ej ← CharBorders(k).R
**4-** **Set** nU ← -1
**5-** **Set** nD ← -1
**6-** **For** all i**Do** {where CharBorders(k).u ≤i≤CharBorders(k).D – 1}
**7-** **For** all j **Do** {where Sj ≤j ≤Ej}
**8-** Check **If** P_DATA(i, j) = 0 **Then**
**9-** Check **If** nU = -1 **Then**
**10-** **Set** nU ← i
**11-** **Else**
**12-** **Set** nD ← i
**13-** **End If**
**14-** **End If**
**15-** **End for**
**16-End for**

**Algorithm (3.7) create character image**

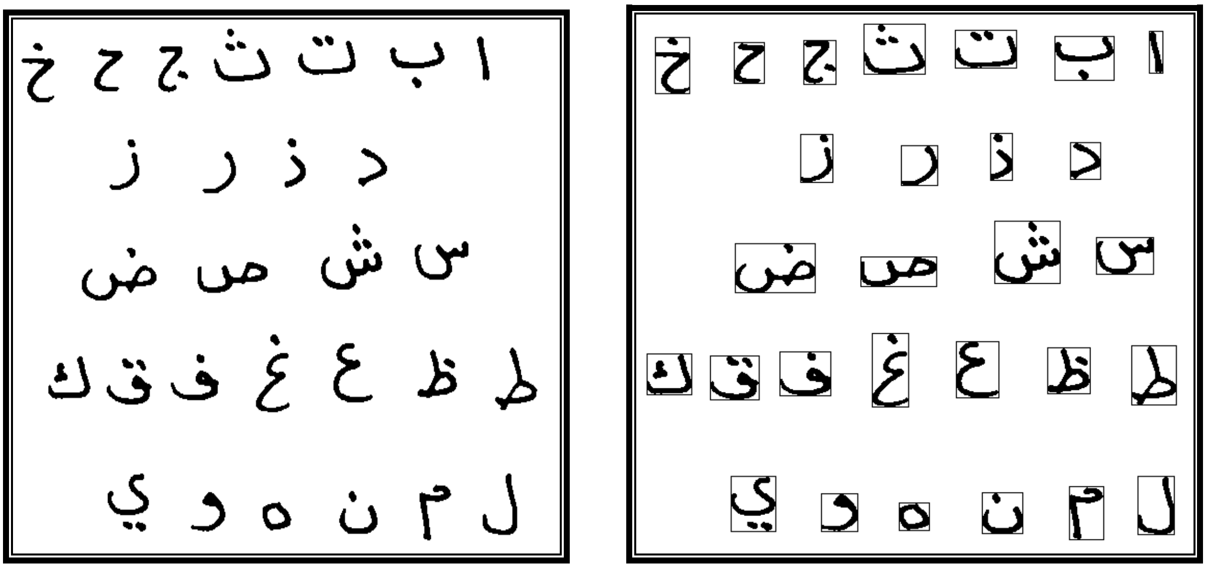
**Input:** Image_Name: name of the image
P_DATA(): image data
H**:** image height
W**:** image width
**Output:** create an image

**1- Set** Header.fsize ← H * W+1078
**2- Set** Header.width ← W
**3- Set** Header.height ← H
**4- Set** Header.imagesize ← H * W
**5-** Open file(give it Image_Name) for binary and print in it the Header values
**6- Set** k ← 0
**7- For** all i,j **Do**{where H - 1 ≥i≥ 0 Step -1 and 0 ≤j≤ W – 1}
**8-** **Set** PIXELS(k) ←P_DATA(i, j)
**9-** **Increment** k by 1
**10-End for**
**11-** save PIXELS on the file and close it
**12-End.**




(a) (b)

Figure (3.6) Image of printed Arabic characters.
(a) Before segmentation process.
(b) After segmentation process.




(a) (b)

Figure (3.7) Image of hand written Arabic characters.
(a) Before segmentation process.
(b) After segmentation process.

The normalization process was important for hand written characters where the characters have different size, so that in this work the window of each character will be ($128\times128$), during the practical this size process the problem of writing the characters in different size.

## 3.6 Feature Extraction (Algorithm and Results)

The set of moments has been shown to be invariant to rotation and scale change. as shown in tables (3.1 to 3.3) the results for images in figure (3.8).

The feature extraction process extracts the features from the training and testing images for all characters (printed). In this work, the character moments (as explained in chapter two) was used as features represents each character, table (3.4) shows features for all training characters, and table (3.5) shows features of testing characters in figure (3.9). To perform the process of features extraction one should perform the following steps:

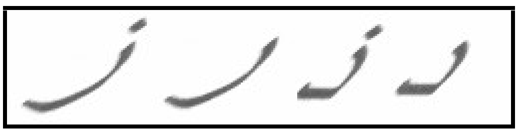
1- Compute ***(mpq)*** value by equation (2.3) in chapter two.

2- Compute ***Central Moment*** value by equation (2.4).

**3-** Compute ***(npq)*** value by equation (2.13)**.**

**4-** Compute moment's values by equations (2.15 - 2.21)**.**


(a)



(b)

(c)

Figure (3.8) (a) Original characters

(b) Rotate the Original characters by $45^0$

(c) Half Size of original characters

Table (3.1) moment invariants for the image in figure (3.8) (a)

| Mom. Order | د | ذ | ر | ز |
|---|---|---|---|---|
| 1 | 7.310902 | 7.308427 | 7.3066 | 7.303604 |
| 2 | 31.80135 | 28.75181 | 30.63034 | 28.1405 |
| 3 | 43.56854 | 39.9241 | 40.41712 | 38.26755 |
| 4 | 33.93864 | 33.1655 | 32.97742 | 32.04677 |
| 5 | 72.50706 | 69.48486 | 69.42756 | 66.77023 |
| 6 | 50.32519 | 47.58441 | 48.39635 | 46.96899 |
| 7 | 71.32059 | 67.45759 | 70.27789 | 66.63087 |

Table (3.2) moment invariants for the image in figure (3.8) (b)

| Mom. Order | د | ذ | ر | ز |
|---|---|---|---|---|
| 1 | 7.311354 | 7.308304 | 7.307242 | 7.304463 |
| 2 | 29.61597 | 28.11388 | 27.2803 | 26.53532 |
| 3 | 40.14907 | 38.63153 | 37.65379 | 36.3426 |
| 4 | 34.15628 | 33.59711 | 33.60384 | 32.66678 |
| 5 | 72.18881 | 71.11051 | 69.71465 | 67.58281 |
| 6 | 49.25549 | 49.32193 | 47.35371 | 46.29332 |
| 7 | 69.64899 | 67.78216 | 68.74391 | 66.72507 |

Table (3.3) moment invariants for the image in figure (3.8) (c)

| Mom. Order | د | ذ | ر | ز |
|---|---|---|---|---|
| 1 | 7.32763 | 7.327291 | 7.32729 | 7.326116 |
| 2 | 35.71868 | 34.21991 | 35.09195 | 34.15151 |
| 3 | 47.23835 | 45.61013 | 45.18116 | 45.21255 |
| 4 | 37.64503 | 37.27551 | 37.31693 | 36.31708 |
| 5 | 79.73325 | 78.47817 | 78.52587 | 76.68049 |
| 6 | 55.64886 | 54.39322 | 55.53623 | 53.73009 |
| 7 | 77.03333 | 75.47568 | 76.23093 | 75.44853 |

Algorithm (3.8) shows how to extract the features for characters. To notice the features of testing handwritten characters see the appendix.

**Algorithm (3.8) Compute_7_Central_Moments**

**Input:** Data for each character

**Output:** Hu's Moment for each character

**1- Set** m00 ← mpq(0, 0)
**2- Set** m10 ← mpq(1, 0)
**3- Set** m01 ← mpq(0, 1)
**4- Set** m20 ← mpq(2, 0)
**5- Set** m02 ← mpq(0, 2)
**6- Set** ib ← m10 / m00
**7- Set** jb ← m01 / m00
**8- Set** N20 ← Npq(ib, jb, 2, 0)
**9- Set** N02 ← Npq(ib, jb, 0, 2)
**10- Set** N11 ← Npq(ib, jb, 1, 1)
**11- Set** N12 ← Npq(ib, jb, 1, 2)
**12- Set** N21 ← Npq(ib, jb, 2, 1)
**13- Set** N30 ← Npq(ib, jb, 3, 0)
**14- Set** N03 ← Npq(ib, jb, 0, 3)
**15-** compute moment's values using equations (2.15 to 2.21) in chapter two for each character
**16-End.**

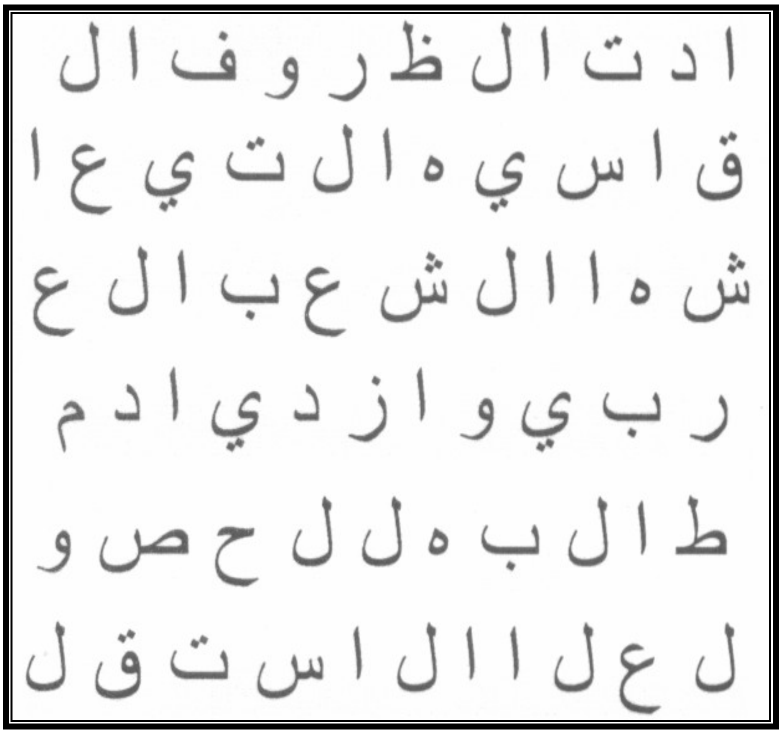



Figure (3.9) Testing printed Arabic characters.

Table (3.4) Features of training characters.

| Character | Mom.1 | Mom.2 | Mom.3 | Mom.4 | Mom.5 | Mom.6 | Mom.7 |
|---|---|---|---|---|---|---|---|
| ا | 7.315547 | 30.57006 | 42.49714 | 37.19798 | 76.72368 | 56.67027 | 78.72356 |
| ب | 7.287325 | 27.581 | 38.02573 | 33.80986 | 69.62821 | 48.18415 | 68.5119 |
| ت | 7.284317 | 29.09393 | 36.50372 | 31.71916 | 65.68723 | 47.27404 | 64.38364 |
| ث | 7.281477 | 29.46027 | 36.20384 | 31.3296 | 65.05502 | 46.58116 | 63.32306 |
| ج | 7.276409 | 27.04892 | 39.389 | 34.86793 | 72.42255 | 48.51801 | 70.6516 |
| ح | 7.2803 | 27.00901 | 39.27912 | 34.58548 | 72.21841 | 48.2348 | 70.07925 |
| خ | 7.277173 | 26.34933 | 36.7227 | 32.9495 | 67.82705 | 46.46526 | 66.83982 |
| د | 7.310902 | 31.80135 | 43.56854 | 33.93864 | 72.50706 | 50.32519 | 71.32059 |
| ذ | 7.308427 | 28.75181 | 39.9241 | 33.1655 | 69.48486 | 47.58441 | 67.45759 |
| ر | 7.3066 | 30.63034 | 40.41712 | 32.97742 | 69.42756 | 48.39635 | 70.27789 |
| ز | 7.303604 | 28.1405 | 38.26755 | 32.04677 | 66.77023 | 46.96899 | 66.63087 |
| س | 7.280565 | 26.61958 | 37.86796 | 33.59595 | 69.27319 | 47.02021 | 71.32785 |
| ش | 7.271932 | 26.61887 | 35.66991 | 30.95939 | 63.90469 | 44.49693 | 64.35136 |
| ص | 7.25324 | 24.20267 | 34.63298 | 31.61171 | 64.73541 | 43.91716 | 64.90322 |
| ض | 7.250596 | 24.49913 | 33.22619 | 30.65544 | 62.29718 | 43.87067 | 62.77864 |
| ط | 7.268446 | 27.11559 | 37.30414 | 30.56212 | 64.43479 | 44.1512 | 61.92638 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ظ | 7.265818 | 27.38254 | 37.55418 | 30.73588 | 64.71629 | 44.43199 | 62.38324 |
| ع | 7.275075 | 26.61097 | 38.1994 | 32.88679 | 68.64079 | 46.40483 | 65.86539 |
| غ | 7.27391 | 25.76097 | 35.56521 | 31.72832 | 65.08688 | 45.28416 | 64.90989 |
| ف | 7.275876 | 27.51961 | 35.51212 | 30.52454 | 63.21819 | 47.41483 | 64.05402 |
| ق | 7.269657 | 27.46378 | 39.34139 | 31.01042 | 65.92944 | 46.70201 | 65.10788 |
| ك | 7.271386 | 29.11914 | 38.46244 | 31.00058 | 65.88887 | 48.17005 | 65.89212 |
| ل | 7.289391 | 27.31222 | 37.06761 | 31.65022 | 65.5159 | 47.37194 | 66.29751 |
| م | 7.303197 | 29.82782 | 44.72677 | 34.20906 | 73.67507 | 49.46453 | 68.69041 |
| ن | 7.29605 | 28.97339 | 41.43269 | 31.99921 | 70.87226 | 46.66278 | 64.7299 |
| ه | 7.310819 | 35.03924 | 46.59023 | 36.03575 | 77.28263 | 54.97655 | 74.95672 |
| و | 7.301809 | 30.69802 | 41.17402 | 33.37029 | 71.59973 | 49.2901 | 68.88318 |
| ي | 7.273513 | 28.24759 | 37.6734 | 33.49772 | 68.72269 | 47.62258 | 70.33869 |

Table (3.5) Features of testing characters.

| Character | Mom.1 | Mom.2 | Mom.3 | Mom.4 | Mom.5 | Mom.6 | Mom.7 |
|---|---|---|---|---|---|---|---|
| ا | 7.315052 | 30.53451 | 43.9119 | 38.76373 | 80.69562 | 54.85365 | 78.79976 |
| د | 7.309571 | 31.42059 | 43.29649 | 33.97339 | 72.12917 | 50.32146 | 72.88552 |
| ت | 7.28509 | 29.94974 | 36.67977 | 31.54067 | 65.56526 | 48.54458 | 63.87173 |
| ا | 7.316143 | 30.71912 | 43.19307 | 37.68117 | 78.12246 | 53.04269 | 77.77497 |
| ل | 7.289474 | 27.35086 | 36.91985 | 31.6053 | 65.36298 | 49.6499 | 65.64728 |
| ظ | 7.266125 | 27.27065 | 37.68729 | 30.70704 | 64.75866 | 44.34803 | 62.29482 |
| ر | 7.308362 | 30.55589 | 40.18925 | 32.87397 | 69.0732 | 49.35492 | 68.54008 |
| و | 7.302472 | 31.00427 | 41.12444 | 33.20076 | 72.34634 | 49.33508 | 67.96004 |
| ف | 7.274006 | 27.12955 | 35.32277 | 30.40901 | 62.90659 | 45.71236 | 64.37602 |
| ا | 7.315761 | 30.76279 | 42.85559 | 37.41795 | 78.83512 | 53.09181 | 75.82462 |
| ل | 7.290824 | 27.28065 | 36.90474 | 31.49815 | 65.23347 | 46.44605 | 66.9651 |
| ق | 7.27255 | 27.72891 | 38.73013 | 31.12678 | 65.76868 | 48.62975 | 64.98486 |
| ا | 7.315885 | 30.71159 | 42.87339 | 38.4632 | 78.63669 | 54.48927 | 78.88544 |
| س | 7.281032 | 26.60277 | 37.79335 | 33.5484 | 69.07627 | 46.91425 | 71.68919 |
| ي | 7.27179 | 28.25557 | 37.45668 | 33.20449 | 68.40941 | 47.43887 | 68.15803 |
| ه | 7.311544 | 34.50494 | 46.80396 | 36.21409 | 78.02625 | 53.72979 | 73.17904 |
| ا | 7.31587 | 30.86615 | 42.9612 | 37.47332 | 77.80198 | 52.92998 | 77.2404 |
| ل | 7.290444 | 27.34537 | 37.1808 | 31.62324 | 65.55937 | 46.98106 | 67.51373 |
| ت | 7.285006 | 29.38002 | 36.55925 | 31.65758 | 65.64095 | 47.49052 | 64.22896 |
| ي | 7.272496 | 28.34184 | 37.49969 | 32.79147 | 67.5498 | 46.96859 | 70.76007 |
| ع | 7.275973 | 26.70132 | 38.16545 | 32.88307 | 68.51619 | 46.49604 | 65.82895 |
| ا | 7.316012 | 30.6659 | 44.56455 | 39.49002 | 81.51479 | 54.83336 | 80.15676 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ش | 7.273046 | 26.78478 | 36.26976 | 31.06542 | 64.38557 | 44.50916 | 67.0413 |
| ه | 7.310935 | 34.05183 | 47.33868 | 36.5404 | 78.25701 | 53.87184 | 74.27302 |
| ا | 7.314806 | 30.59298 | 42.81021 | 38.22984 | 78.33806 | 56.35951 | 80.16183 |
| ا | 7.31518 | 30.5757 | 42.54584 | 37.43477 | 77.77792 | 53.48029 | 76.06542 |
| ل | 7.291338 | 27.53609 | 37.22754 | 31.75475 | 65.75175 | 48.25137 | 66.37212 |
| ش | 7.271688 | 26.56732 | 35.99447 | 31.0943 | 64.22614 | 44.44336 | 65.6887 |
| ع | 7.276769 | 26.7761 | 38.03354 | 32.92843 | 68.82313 | 46.46898 | 66.02644 |
| ب | 7.291436 | 27.68238 | 38.35828 | 34.14996 | 70.36741 | 49.33504 | 70.05171 |
| ا | 7.315044 | 30.59969 | 42.96345 | 38.06643 | 78.09097 | 53.83235 | 78.66727 |
| ل | 7.29012 | 27.4426 | 37.04663 | 31.51287 | 65.31212 | 46.69518 | 66.63873 |
| ع | 7.277397 | 26.78259 | 38.09958 | 32.91035 | 68.80809 | 46.45387 | 65.97582 |
| ر | 7.307889 | 31.03763 | 40.52057 | 33.01797 | 69.49141 | 48.68303 | 69.78265 |
| ب | 7.2915 | 27.7155 | 38.54965 | 34.13734 | 70.50967 | 50.17615 | 70.97394 |
| ي | 7.274566 | 28.3357 | 37.5204 | 33.34752 | 68.68259 | 47.64973 | 68.4059 |
| و | 7.302609 | 30.97764 | 41.19164 | 33.21387 | 73.56021 | 49.16813 | 67.70035 |
| ا | 7.316364 | 30.80505 | 43.5839 | 39.18226 | 80.05843 | 55.41617 | 80.27072 |
| ز | 7.305167 | 28.3279 | 38.54347 | 32.22211 | 67.10851 | 47.38827 | 66.88255 |
| د | 7.30996 | 31.52589 | 43.12843 | 33.7917 | 71.9737 | 50.12367 | 71.32493 |
| ي | 7.273773 | 28.36497 | 37.74887 | 33.03681 | 67.96107 | 47.22887 | 68.35756 |
| ا | 7.315798 | 30.49371 | 42.74019 | 37.73256 | 77.67677 | 56.30762 | 79.03942 |
| د | 7.309712 | 31.65745 | 42.73422 | 33.85299 | 71.79581 | 50.39554 | 70.75356 |
| م | 7.304129 | 29.96684 | 47.11253 | 34.44175 | 75.28621 | 49.69753 | 69.1264 |
| ط | 7.274066 | 27.28137 | 37.89334 | 30.78301 | 65.05178 | 44.46235 | 62.24684 |
| ا | 7.31543 | 30.53993 | 42.69145 | 37.48087 | 77.44336 | 54.3394 | 76.95423 |
| ل | 7.293434 | 27.31775 | 37.12341 | 31.76424 | 65.73362 | 47.05724 | 67.19461 |
| ب | 7.291357 | 27.67867 | 38.16669 | 34.16274 | 70.22455 | 48.98868 | 69.63411 |
| ه | 7.310843 | 33.9859 | 45.63031 | 35.6599 | 75.89587 | 53.12046 | 74.28146 |
| ل | 7.292052 | 27.37726 | 37.14486 | 31.68848 | 65.619 | 47.18809 | 66.76593 |
| ل | 7.290809 | 27.31423 | 36.95712 | 31.55468 | 65.33666 | 46.5776 | 67.68321 |
| ح | 7.280483 | 27.09519 | 39.44921 | 34.47123 | 72.68986 | 48.58968 | 69.90612 |
| ص | 7.25589 | 24.34023 | 34.92989 | 31.80804 | 65.17709 | 44.2037 | 65.15181 |
| و | 7.30271 | 31.03027 | 41.20455 | 33.17008 | 71.93522 | 49.40748 | 67.97345 |
| ل | 7.291834 | 27.36832 | 37.11161 | 31.7105 | 65.65437 | 46.99118 | 68.4278 |
| ع | 7.277407 | 26.64127 | 37.98772 | 32.86257 | 68.03482 | 46.71885 | 65.74265 |
| ل | 7.291398 | 27.48231 | 37.0247 | 31.60793 | 65.42451 | 47.31362 | 65.92627 |
| ا | 7.315906 | 30.64406 | 43.68723 | 38.62089 | 80.90884 | 54.46503 | 78.25464 |
| ا | 7.316496 | 30.7776 | 43.16426 | 37.73681 | 78.97952 | 53.32146 | 76.52579 |
| ل | 7.291983 | 27.42805 | 37.20428 | 31.70469 | 65.67756 | 47.13992 | 67.27256 |
| ا | 7.31612 | 30.79373 | 42.63544 | 37.50579 | 77.18394 | 53.10694 | 77.4116 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| س | 7.282094 | 26.66718 | 37.66631 | 33.49126 | 68.8778 | 46.83265 | 69.17411 |
| ت | 7.289097 | 29.34102 | 36.63251 | 31.86473 | 65.98357 | 47.24487 | 64.54713 |
| ق | 7.273012 | 27.69134 | 39.10685 | 31.00586 | 65.58984 | 47.15555 | 64.76826 |
| ل | 7.292685 | 27.4032 | 37.2118 | 31.75657 | 65.75807 | 47.48087 | 66.74925 |

## 3.7- Pattern Matching

In the pattern matching process, the Euclidean distance used to show the percentage accuracy of matching between training and testing characters, as shown in equation (2.22). The matching process was performed between the features extracted from training characters and the testing characters. Algorithm (3.9) shows how to perform the pattern matching process.

**Algorithm (3.9) Pattern Matching**

**Input:** Features of (testing and training ) characters

**Output:** Percentage accuracy.
Display the best matched characters onto a ListBox

**1**- Read CharImage from files,
**2-** Extract its H & W with Char_DATA
**3-** Compute the 7 central moments as a Feature Extraction Method on Char_DATA
**4- If** Training **then** write features on Training_File
**5- Else**
**6- If** Testing **then**
**7-** Extract Features onto a 1-D array and Compare it with the Training File
**8-** Find the best matched character and compare it with the Test_Texti.txt file
**9-** Compute the percentage accuracy
**10-** Display the Recognized charcters onto a ListBox
**11- Else**
**12- If** OCR **then**
**13-** Extract Features onto a 1-D array and Compare it with the Training File
**14-** Display the best matched characters onto a ListBox
**15- End if**
**16- End.**

## 3.8 Recognition Decision

The recognition decision ratio for some characters contained in one image depends on the following:

- ► Number of characters in the testing image.
- ► Type of characters in the testing image.
- ► How many times the same character was repeated.
- ► The order of moments that are used for matching.
- ►The text type (i.e. printed characters or hand-written characters).

Two samples of images are tested in this work, the first one is printed Arabic characters and the second is hand written characters.

### *1- printed characters:*

In this stage the image of printed characters shown in Figure (3.9) contains ***65*** characters, some of these characters are repeated like (ل، ع، ر). When the characters in this image are tested, It was found that when all the orders of moments are used the percentage accuracy equal (87.692307%), while when the (second order, third order and fourth order) of moments are used, the percentage accuracy became (96.923076%). Table (3.6) proves that the (second, third and fourth) orders of moment giving good results than the other orders. Table (3.7) shows the recognition of characters using all orders of moments and recognition of characters when the specific orders of moments are used. In this table the characters in shading cells are not recognized correctly.

Table (3.6) Recognition results using specific orders of moments.

| Moments order | Percentage accuracy |
|---|---|
| $1^{st}$ | 50.76923% |
| $2^{nd}$ | 44.61538% |
| $3^{rd}$ | 50.76923% |
| $4^{th}$ | 46.15384% |
| $5^{th}$ | 21.53846% |
| $6^{th}$ | 23.07692% |
| $7^{th}$ | 33.84615% |
| $1^{st},2^{nd},3^{rd},4^{th},5^{th},6^{th},7^{th}$ | 87.6923% |
| $1^{st},2^{nd},3^{rd},4^{th},5^{th},6^{th}$ | 92.3076% |
| $1^{st},2^{nd},3^{rd},4^{th},5^{th}, 7^{th}$ | 87.6923% |
| $1^{st},2^{nd},3^{rd},4^{th}, 6^{th},7^{th}$ | 83.0769% |
| $1^{st},2^{nd},3^{rd}, 5^{th},6^{th},7^{th}$ | 86.1538% |
| $1^{st},2^{nd}, 4^{th},5^{th},6^{th},7^{th}$ | 84.6153% |
| $1^{st}, 3^{rd},4^{th},5^{th},6^{th},7^{th}$ | 83.0769% |
| $2^{nd},3^{rd},4^{th},5^{th},6^{th},7^{th}$ | 87.6923% |
| $2^{nd}, 3^{rd}, 4^{th}$ | 96.92307% |
| $5^{th},6^{th},7^{th}$ | 61.53846% |
| $1^{st},2^{nd}, 3^{rd}, 4^{th}$ | 96.92307% |
| $1^{st}$., $2^{nd}$. | 46.15384% |
| $1^{st}, 3^{rd}$ | 50.76923% |
| $1^{st}, 4^{th}$ | 46.15384% |
| $2^{nd}, 3^{rd}$ | 83.07692% |
| $2^{nd}, 4^{th}$ | 92.30769% |
| $3^{rd}$., $4^{th}$. | 89.23076% |
| $5^{th},6^{th}$ | 38.46153% |
| $5^{th},7^{th}$ | 69.23076% |
| $6^{th},7^{th}$ | 40% |

| | |
|---|---|
| 1st,2nd,3rd,4th,5th | 93.8461% |
| 1st,2nd,3rd,4th,6th | 93.8461% |
| 1st,2nd,3rd,4th,7th | 83.0769% |
| 1st, 2nd, 3rd | 83.07692% |
| 1st, 2nd, 4th | 92.30769% |
| 1st, 3rd, 4th | 89.23076% |
| 1st, 5th, 6th, 7th | 61.5384% |
| 1st, 4th, 5th, 6th | 46.1538% |
| 4th, 5th, 6th, 7th | 64.6153% |
| 1st, 2nd, 5th | 75.3846% |
| 1st, 3rd, 6th, 7th | 66.1538% |
| 1st, 2nd, 5th, 7th | 80% |
| 1st, 2nd, 5th, 6th | 80% |

Table (3.7) Recognition decision of printed characters.

| Input character | Recognized by 7-Mom. | Recognized by 2nd, 3rd, and 4th Mom. |
|---|---|---|
| ا | ا | ا |
| د | د | د |
| ت | ت | ث |
| ا | ا | ا |
| ل | ل | ل |
| ظ | ظ | ظ |
| ر | ر | ر |
| و | و | و |
| ف | ش | ف |
| ا | ا | ا |
| ل | ل | ل |

| | | |
|---|---|---|
| ق | ك | ق |
| ا | ا | ا |
| س | س | س |
| ي | ب | ي |
| ه | ه | ه |
| ا | ا | ا |
| ل | ل | ل |
| ت | ت | ت |
| ي | ي | ي |
| ع | ع | ع |
| ا | ا | ا |
| ش | غ | ش |
| ه | ه | ه |
| ا | ا | ا |
| ا | ا | ا |
| ل | ل | ل |
| ش | ش | ش |
| ع | ع | ع |
| ب | ب | ب |
| ا | ا | ا |
| ل | ل | ل |
| ع | ع | ع |
| ر | ر | ر |
| ب | ج | ب |
| ي | ب | ي |
| و | و | و |
| ا | ا | ا |

| | | |
|---|---|---|
| ز | ز | ز |
| د | د | د |
| ي | ي | ي |
| ا | ا | ا |
| د | د | د |
| م | م | م |
| ط | ظ | ظ |
| ا | ا | ا |
| ل | ل | ل |
| ب | ب | ب |
| ه | ه | ه |
| ل | ل | ل |
| ل | ل | ل |
| ح | ح | ح |
| ص | ص | ص |
| و | و | و |
| ل | ل | ل |
| ع | ع | ع |
| ل | ل | ل |
| ا | ا | ا |
| ا | ا | ا |
| ل | ل | ل |
| ا | ا | ا |
| س | ب | س |
| ت | ت | ت |
| ق | ق | ق |
| ل | ل | ل |

## *2- hand-written characters:*

In this phase:

- The characters written by type of font that are most of peoples used it in most times which is .
- Characters that are tested written in different way by different persons using pen of type (Soft) for writing on white paper.
- The (second, third and fourth) orders of moments are used for matching, because it was given height percentage accuracy than the other orders.

In this work some characters are tested and gives different percentage accuracy, as shown in tables (3.8-3.13). In this table the characters in the shading cells are not recognized correctly, this belongs to the different in the size and shape of the same character and the training characters that the test characters matched with it considered to be standard shape (printed characters).

Table (3.8) Recognition results of character (ف) written by different writers.

| **Writers** | **Input character** | **Rec. Results** |
|---|---|---|
| **1** |  | ف |
| **2** |  | ض |
| **3** |  | ث |
| **4** |  | ف |
| **5** |  | ض |

Table (3.9) Recognition results of character (ج) written by different writers.

| **Writers** | **Input character** | **Rec. Results** |
|---|---|---|
| **1** |  | ج |
| **2** |  | ع |
| **3** |  | ج |
| **4** |  | ذ |
| **5** |  | ب |

Table (3.10) Recognition results of character (غ) written by different writers.

| **Writers** | **Input character** | **Rec. Results** |
|---|---|---|
| **1** |  | ص |
| **2** |  | ص |
| **3** |  | ص |
| **4** |  | ض |
| **5** |  | ص |

Table (3.11) Recognition results of character (ض) written by different writers.

| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ض |
| 2 |  | ض |
| 3 |  | ض |
| 4 |  | ض |
| 5 |  | ض |

Table (3.12) Recognition results of character (ظ) written by different writers.

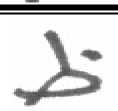
| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ش |
| 2 |  | ل |
| 3 |  | ش |
| 4 |  | ش |
| 5 |  | غ |

Table (3.13) Recognition results of character (ز) written by different writers.

| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ز |
| 2 |  | ز |
| 3 |  | ز |
| 4 |  | ع |
| 5 |  | ل |

Note that, in appendix more tested of hand written characters.

## 3.9 Discussion

It was very interested to find the most useable parameter among adopted order of moments, since the use of all them sometimes indicates some weakness in the recognition decision. This prepares to search on the mono-directional moments that share with each other to strength the recognition ratio.

Table (3.6) shows the use of the all moment orders for the input characters shown in Figure (3.9). It is noticeable that when each order was used as alone, the percentage accuracy of first fourth orders higher than the percentage accuracy of last three orders of moments. Also when used all orders with each other the percentage accuracy was (87.6923%), it was note that affect each order on the other as following: remove the 7$^{th}$ Order the result was increased, remove the 6$^{th}$ Order not affect on the results, remove (5$^{th}$, 4$^{th}$, 3$^{rd}$, 2$^{nd}$) orders the results was decreased and remove the 1$^{st}$ not affect on the results.

While the first order moment show very small variation and it was not affect when it was shared with other moments in the recognition task, this indicates it is useless individually, so that it can not be employed in the recognition. During the practical the percentage accuracy of coalition work of the ($2^{nd}$, $3^{rd}$ and $4^{th}$) higher than the resulted moments of orders ($5^{th}$, $6^{th}$ and $7^{th}$) which appears less recognition percentage in comparison with other orders. Also when different orders of moments used with each other as mentioned in the results shown in table (3.6) all of these lead us to one result which is the ($2^{nd}$, $3^{rd}$ and $4^{th}$) giving higher percentage accuracy which improve that the second moment is a measure of gray-level contrast that can be used to establish descriptors of relative smoothness, the third moment is a measure of the skewness of the histogram while the fourth moment is a measure of its relative flatness, the fifth and higher moments are not so easily related to character shape. From all of this handwritten characters recognized using ($2^{nd}$, $3^{rd}$ and $4^{th}$) which are giving good results.

# Chapter Four

# Conclusions and Future Work

# Chapter Four
# Conclusions and Future Work

## 4.1 conclusions

From the current work, it can be concluded that:

► Hu's seven moments matching is an efficient technique for printed Arabic character recognition. Recognition system has been performed on the printed Arabic characters and the percentage accuracy for recognize 65 printed characters was 96.92307%, while in hand written characters, the recognition percentage is decreased due to irregularities appears in the handwritten characters.

► It was found that the variation in the first order moments was very small, so that it can not be employed in the recognition. While the second order moments, third order moment and fourth order moments giving higher recognition rate than other order moments.

► The fifth and higher moments are not so easily related to character recognition.

## 4.2 Future Work

Some important proposed aspects for future investigations are given:

♦ The order of moments is sensitive to style where it was depends on the spatial information, therefore future work is required to make investigation in the relationships between the order of moment and the style of the character.

♦ The gradient of moments affects by the noise and deformation, therefore, it can be studied the relationship between the gradient of moments with the noise and deformation.

♦ Perform edge thinning which it is an attempt to represent the structural shape of a plane region and reduce it to a graph; this reduction may be accomplished by obtaining the skeleton of the region via a thinning algorithm. This step will reduce some of the variance that is usually found in characters like the width of their edges, therefore, it considered as a uniforming process to make all the characters with 1 pixel width.

♦ Although this research thesis develops a recognizer of Arabic text with a certain type of font style, can be add other types of fonts and styles and support the user with access to interactively deciding which font in the existing library is nearest before starting the recognition task.

# References

# References

●**[Adi91]** L. Adisleven, F. Miriam, "Recognition of Handwritten Hebrew One Stroke Letters by Learning Syntactic Representations of Symbols", IEEE Trans, Sys_Man and Cybernetics,Vol. 19, No. 5, pp. 1306 – 1313, 1991.

● **[Boz89]** R. M. Bozinovic, S. N. Srihari "Off-line Cursive Script Word Recognition", IEEE Trans, Pattern recognition and Machine Intelligence, Vol. 11, 1989.

● **[Cas95]** R. G. Casey, E. Lecolinet, "Strategies in character segmentation: A survey", in Proceedings of International Conference on Document Analysis and Recognition, pp.1028-1033, 1995.

● **[Dab90]** S. Dabi, R. Ramsis and A. Kamel, "Arabic Character Recognition System: Statistical Approach for Recognizing Cursive Typewritten Text", Pattern Recognition Recoion J. Vol. 23, pp. 485-495, 1990.

● **[Dev96]** L. Devroye, L. Gyorfi and G. Lugosi, "A Probabilistic Theory of Pattern Recognition", Berlin: Springer Verlag, Vol. 12, 1996.

● **[Fu82]** K. S. Fu, "Syntactic Pattern Recognition and Applications", Englewood Cliffs, N.J., Prentice-Hall, 1982.

● **[Gov90]** V. K. Govindan, A. P. Shivaprasad, " Character recognition", A review", Pattern recognition Recoion J. Vol. 23, No. 7, pp. 671-683, 1990.

● **[Gon78]** R. C. Gonzalez, M. G. Thomason, "Syntactic Pattern Recognition, An Introduction", Addison Wesley, London 1978.

● **[Hu62]** M. K. Hu, “Visual Pattern Recognition by Moment Invariants”, IEEE Trans. Infor. Theory Vol. 8, pp. 179-187, 1962.

● **[Hua79]** T. S. Huang, G. I. Yang and G. Y. Tang, “A Fast Two-Dimensional Median Filtering Algorithm” IEEE Trans. Acoustics Speech and Signal Processing, ASSP Vol. 27, No. 1, pp. 13-18, February 1979.

● **[Jai95]** R. Jain, R. Kasturi, B. G. Schunk, "Machine Vision", McGraw-Hill, New York, 1995.

● **[Kam02]** C. Kam-Fai, Y. Dit-Yan, "An effecient syntactic approach to structural analysis of on-line handwritten mathematical expressions", Pattern Recognition, Vol. 33, pp. 375-384, 2002.

● **[Kep90]** C. Keping, "Efficient Parallel Algorithms for The Computation of Two Dimensional Image Moments", Pattern Recognition, Vol. 21, No. 2, pp. 109 – 119, 1990.

● **[Lei07]** S. Leishman, "Shape-Free Statistical Information in Optical Character Recognition" , M.Sc. thesis, University of Toronto, 2007.

●**[Mar92]** V. Margner, “SARAT - A System for the Recognition of Arabic Printed Text”, Proc. 11th Int.Conf. On Pattern Recognition, pp. 561-564, 1992.

●**[Mar05]** S. Marukatat, S. Sae-tang and T. Theeramunkung, “The feature Combination Technique for Off-line Thai Character Recognition System”, Eighth International Conference on Document Analysis and Recognition (ICDAR), Vol. 05, pp. 1006-1009, 2005.

●**[Mer05]** M. Mercimek, K. Gulez, T. Velimumcu, "Real objects recognition using moment invariants", Besiktas-Istanbul, Turkey, April 2005.

● **[Moh95]** E.J. Mohamed., "Arabic Character Recognition Methodology using Complex Moment", M.Sc. thesis, University of Technology 1995.

● **[Naf98]** A. Nafiz "off_line character recognition system for free style handwriting ", M.Sc. thesis, Middle East Technical University 1998.

● **[Pav75]** T. Pavilideis, F. Ali, "Computer Recognition of Handwritten Numerals by Polygonal Approximations", IEEE Trans. Comput., pp. 610 – 614, 1975.

● **[Qin03]** C. Qing, M. Emil "Optical Character Recognition for Model-based Object Recognition Applications ", Petriu University of Ottawa, Ottawa, On, Canada 2003.

●**[Rob00]** P. W. Robert, J. Mao "Statistical Pattern Recognition" IEEE trans. pattern analysis and machine intelligence, Vol. 22, No. 1, january 2000.

● **[Rob01]** T. O. Robert, “Generalized Feature Extraction for Structural Pattern Recognition in Time Series Data”, PhD thesis, University-Pittsburgh, 2001.

● **[Rot76]** F. M. Roth, "Representation of Structural Events and Efficient Procedures for their Recognition", Pattern Recognition, Vol. 8, pp. 141-150, 1976.

●**[Sue80]** C. Y. Suen, M. Berthod and S. Mori, "Automatic Recognition of Hand Printed Characters", the State of the Art in Proceedings of the IEEE, Vol. 68, No.4, pp.469 – 487, 1980.

● **[Sna04]** B. Snakur, M. Sezgin, “Thresholding techniques and quantitative performance. evaluation”, Journal Electron. Imaging, Vol. 13, No.1, pp.146-165, 2004.

●**[Taa05]** Ahmad T. Al-Taani "An Efficient Feature Extraction Algorithm for the Recognition of Handwritten Arabic Digits" International Journal of Computational Intelligence Vol. 2, No. 2, ISSN 1304-4508, 2005.

● **[Tho02]** G. Thomas, "Machine Learning", Department of Computer Science, Oregon State University Corvallis, OR 97331, 2002

●**[Tia91]** Q. Tian, P. Zhang, T. Alexander and Y. Kim, "Survey_ Omnifont Printed Character Recognition_ Visual Communication and Image Processing", Image Processing, pp. 260 – 268, 1991.

● **[Tim01]** K. Tim "Towards Neural Network Recognition Of Handwritten Arabic Letters" ,M.Sc. thesis, Dalhousie University 2001.

● **[Tin05]** A. Tinku, "Image Processing ", Published by John Wiley & Sons, Inc., Hoboken, New Jersey. Published simultaneously in Canada 2005.

● **[Ula87]** N. Ula “A Variable Limb-Width Matching Algorithm for Automatic Recognition of Arabic Characters”, the 10th Symposium on Information Theory and it's Applications, pp. 411-415, 1987.

● **[Zha02]** D. Zhang and G. Lu, “A comparative study of three region shape descriptors”, Digital Image Computing Techniques and Applications, pp. 21-22, Melbourne Australia, January 2002.

# Appendix

Table (1) Features of testing character ف

| Character | Mom1 | Mom2 | Mom3 | Mom4 | Mom5 | Mom6 | Mom7 |
|---|---|---|---|---|---|---|---|
| ف | 7.267456 | 27.4133 | 34.97936 | 30.05857 | 62.30326 | 43.80428 | 63.40422 |
| ف | 7.263379 | 26.17107 | 33.41296 | 29.71725 | 60.94698 | 43.44284 | 62.08093 |
| ف | 7.265194 | 30.54745 | 35.70778 | 30.23279 | 62.82766 | 45.53017 | 64.90718 |
| ف | 7.259233 | 27.40888 | 34.6127 | 29.93696 | 62.01619 | 44.14101 | 63.88693 |
| ف | 7.249269 | 25.17774 | 32.97499 | 29.34661 | 60.18499 | 42.60125 | 61.19062 |
| ف | 7.260846 | 27.54739 | 34.66724 | 30.25289 | 62.34188 | 44.97364 | 63.50227 |
| ف | 7.250003 | 25.84065 | 34.10297 | 29.9977 | 61.70858 | 44.51361 | 63.00478 |
| ف | 7.265599 | 28.15552 | 35.7021 | 30.42297 | 63.14661 | 45.2205 | 63.40263 |
| ف | 7.254569 | 26.79295 | 34.19987 | 29.72202 | 61.27252 | 43.72283 | 61.89961 |
| ف | 7.272798 | 26.93312 | 34.99648 | 30.25161 | 62.56841 | 44.36965 | 62.11423 |
| ف | 7.269261 | 27.17932 | 35.527 | 30.75059 | 63.53622 | 45.03674 | 63.47112 |
| ف | 7.267631 | 27.59413 | 35.16519 | 30.32687 | 62.80112 | 44.50374 | 62.17427 |
| ف | 7.25615 | 27.39744 | 34.61882 | 30.13373 | 62.14133 | 43.89693 | 62.64175 |
| ف | 7.256061 | 25.95361 | 33.96564 | 29.61565 | 61.0704 | 43.11981 | 60.93383 |
| ف | 7.248302 | 25.95785 | 33.85144 | 29.30854 | 60.61335 | 42.51978 | 60.11628 |
| ف | 7.262079 | 26.9438 | 34.76558 | 30.09042 | 62.39341 | 44.47845 | 61.48408 |
| ف | 7.268048 | 29.09233 | 36.21772 | 31.05225 | 64.28249 | 45.66378 | 65.25107 |
| ف | 7.260115 | 27.49269 | 35.35398 | 29.98803 | 62.36049 | 44.02205 | 61.57447 |
| ف | 7.276576 | 28.14089 | 35.6581 | 30.49489 | 63.31253 | 44.78516 | 64.07693 |
| ف | 7.267062 | 28.10091 | 35.40265 | 30.3011 | 62.74372 | 44.81005 | 63.55304 |

Table (2) Features of testing characterح

| Character | Mom1 | Mom2 | Mom3 | Mom4 | Mom5 | Mom6 | Mom7 |
|---|---|---|---|---|---|---|---|
| ح | 7.270134 | 27.08948 | 38.38297 | 35.28392 | 72.11476 | 48.8361 | 72.56828 |
| ح | 7.253691 | 27.18406 | 37.86198 | 32.39317 | 67.38271 | 46.63436 | 69.78333 |
| ح | 7.275457 | 27.6122 | 40.83363 | 36.14824 | 75.73351 | 49.95447 | 73.25936 |
| ح | 7.215377 | 25.02766 | 36.6615 | 31.4788 | 67.59609 | 44.04078 | 63.20394 |
| ح | 7.278899 | 27.90201 | 38.10574 | 34.10518 | 70.77201 | 48.26597 | 69.61765 |
| ح | 7.279694 | 27.93991 | 39.84371 | 34.80342 | 72.43327 | 49.60419 | 72.37795 |
| ح | 7.279984 | 28.47687 | 39.36253 | 34.75826 | 72.0565 | 49.07512 | 70.22581 |
| ح | 7.272771 | 28.3493 | 38.71453 | 33.58543 | 69.60824 | 49.77849 | 67.23228 |
| ح | 7.281943 | 27.96092 | 41.0927 | 37.48083 | 77.50507 | 54.26484 | 75.15917 |
| ح | 7.27109 | 26.45272 | 42.61427 | 38.10927 | 79.29336 | 51.44321 | 78.37408 |
| ح | 7.275282 | 27.73526 | 40.0531 | 35.16326 | 72.94357 | 49.74523 | 70.56512 |
| ح | 7.254004 | 27.17612 | 39.4015 | 35.19215 | 72.72613 | 49.44176 | 71.91148 |
| ح | 7.260358 | 26.61211 | 40.00121 | 34.74553 | 74.2571 | 51.84527 | 69.5179 |
| ح | 7.262239 | 26.73897 | 37.55745 | 35.78946 | 73.08768 | 49.28189 | 72.84868 |
| ح | 7.266705 | 27.35126 | 42.54261 | 34.71534 | 75.44127 | 50.83182 | 73.26151 |
| ح | 7.263629 | 26.72926 | 37.7015 | 38.47754 | 76.31509 | 52.47021 | 76.89263 |
| ح | 7.267468 | 27.05986 | 38.35966 | 39.93016 | 79.42526 | 53.77988 | 80.51699 |
| ح | 7.262506 | 26.29485 | 38.96204 | 37.66334 | 76.22057 | 50.84541 | 76.72264 |
| ح | 7.259173 | 26.49525 | 38.98932 | 36.0447 | 73.32342 | 49.7367 | 73.50986 |
| ح | 7.26231 | 27.29288 | 37.97223 | 33.52653 | 69.92023 | 47.17297 | 67.93943 |

Table (3) Features of testing character غ

| Character | Mom1 | Mom2 | Mom3 | Mom4 | Mom5 | Mom6 | Mom7 |
|---|---|---|---|---|---|---|---|
| غ | **7.255045** | **23.59429** | **34.9576** | **30.23007** | **62.56033** | **42.88426** | **60.52436** |
| غ | **7.252999** | **23.6642** | **34.74081** | **32.52503** | **66.08889** | **44.35739** | **65.26826** |
| غ | **7.265438** | **24.73791** | **35.00724** | **31.35401** | **64.2076** | **44.1181** | **64.85191** |
| غ | **7.236696** | **24.01401** | **34.28245** | **30.10908** | **62.05503** | **42.13548** | **60.60546** |
| غ | **7.260639** | **24.50221** | **35.15899** | **31.70546** | **64.68169** | **44.19545** | **64.5191** |
| غ | **7.264663** | **24.43693** | **34.77587** | **31.83372** | **65.09769** | **44.20488** | **64.26334** |
| غ | **7.262594** | **24.83584** | **35.42825** | **31.48049** | **65.07832** | **44.02345** | **63.6087** |
| غ | **7.262774** | **24.50266** | **35.83567** | **31.09702** | **64.12254** | **46.36236** | **62.53931** |
| غ | **7.263336** | **24.61634** | **34.16856** | **30.90525** | **63.87553** | **45.93592** | **62.04815** |
| غ | **7.261582** | **24.29068** | **33.92847** | **31.46744** | **63.84463** | **43.69683** | **63.41045** |
| غ | **7.265109** | **24.95131** | **34.20251** | **31.0099** | **63.76348** | **44.37444** | **62.17999** |
| غ | **7.229713** | **23.51615** | **33.07572** | **29.89048** | **61.07254** | **44.69715** | **60.12835** |
| غ | **7.269408** | **25.14984** | **36.0046** | **32.40638** | **66.12852** | **45.3979** | **66.41096** |
| غ | **7.265197** | **25.52377** | **36.40493** | **32.26455** | **66.41949** | **45.76414** | **66.81631** |
| غ | **7.258098** | **24.33825** | **34.45073** | **31.53218** | **64.42033** | **46.23174** | **65.20253** |
| غ | **7.264865** | **24.94021** | **34.95721** | **30.91506** | **63.50081** | **45.66644** | **62.306** |
| غ | **7.26719** | **25.25565** | **36.54685** | **32.26423** | **66.31799** | **45.35825** | **65.92751** |
| غ | **7.264361** | **24.71477** | **35.6635** | **31.69763** | **65.48191** | **44.06776** | **63.59671** |
| غ | **7.259548** | **24.23328** | **35.70295** | **30.75895** | **63.52402** | **43.51096** | **63.28497** |
| غ | **7.258033** | **24.28134** | **35.85031** | **32.07919** | **65.56995** | **44.59188** | **65.61314** |

Table (4) Features of testing character ض

| Character | Mom1 | Mom2 | Mom3 | Mom4 | Mom5 | Mom6 | Mom7 |
|---|---|---|---|---|---|---|---|
| ض | 7.240337 | 24.45023 | 33.40693 | 30.08277 | 61.43432 | 42.70626 | 61.61219 |
| ض | 7.252912 | 25.41212 | 34.10235 | 30.31265 | 62.01899 | 43.12695 | 63.20728 |
| ض | 7.247306 | 24.22751 | 33.23646 | 30.28592 | 61.61114 | 42.47878 | 61.85094 |
| ض | 7.236795 | 24.00864 | 32.98504 | 30.42384 | 61.62008 | 42.77453 | 62.53468 |
| ض | 7.217194 | 23.52951 | 32.27944 | 29.10559 | 59.3708 | 43.37978 | 59.99991 |
| ض | 7.212417 | 24.0707 | 34.26954 | 30.77245 | 62.83815 | 43.23184 | 64.14156 |
| ض | 7.236681 | 24.62763 | 34.96376 | 30.68873 | 63.00854 | 43.28237 | 64.27159 |
| ض | 7.241087 | 23.8127 | 33.36235 | 30.37667 | 61.75571 | 42.67226 | 62.42562 |
| ض | 7.246949 | 24.57477 | 33.07265 | 30.01887 | 61.07294 | 42.50974 | 61.75477 |
| ض | 7.230438 | 24.13666 | 33.14514 | 29.79827 | 60.99313 | 41.9594 | 60.53856 |
| ض | 7.222189 | 23.38451 | 32.7723 | 29.83379 | 60.81097 | 41.6988 | 60.85381 |
| ض | 7.238925 | 25.12441 | 33.97635 | 30.52398 | 62.57419 | 43.08815 | 61.6599 |
| ض | 7.240071 | 24.87186 | 33.83382 | 30.12428 | 61.66298 | 42.78664 | 62.09481 |
| ض | 7.24509 | 25.06778 | 33.99133 | 30.30669 | 62.01156 | 42.87181 | 62.28984 |
| ض | 7.247884 | 25.24223 | 33.41246 | 29.92957 | 61.255 | 42.60424 | 60.87848 |
| ض | 7.249551 | 25.49987 | 33.91709 | 30.15302 | 61.84666 | 42.97793 | 61.44394 |
| ض | 7.246674 | 24.76364 | 33.29547 | 30.28261 | 61.65627 | 42.73608 | 61.7688 |
| ض | 7.246124 | 24.49284 | 33.54463 | 30.56232 | 62.31134 | 42.90605 | 61.99461 |
| ض | 7.244448 | 24.44227 | 33.84941 | 30.42333 | 62.11193 | 43.16319 | 62.65806 |
| ض | 7.247792 | 24.6903 | 33.50443 | 30.01571 | 61.38631 | 42.44442 | 61.24995 |

Table (5) Features of testing character ظ

| Character | Mom1 | Mom2 | Mom3 | Mom4 | Mom5 | Mom6 | Mom7 |
|---|---|---|---|---|---|---|---|
| ظ | 7.240451 | 26.93324 | 36.15723 | 30.14299 | 62.94811 | 45.37836 | 64.36704 |
| ظ | 7.228362 | 25.58582 | 35.08138 | 29.08587 | 60.76543 | 42.2431 | 59.78022 |
| ظ | 7.218229 | 25.57267 | 34.82479 | 29.54923 | 61.37063 | 42.79049 | 62.33468 |
| ظ | 7.257565 | 26.71207 | 36.42996 | 30.7477 | 64.10979 | 44.87122 | 64.16389 |
| ظ | 7.255772 | 25.91539 | 35.31841 | 31.14152 | 64.20634 | 46.94631 | 64.48306 |
| ظ | 7.255526 | 25.87463 | 34.10555 | 30.71091 | 63.66215 | 43.82262 | 61.86797 |
| ظ | 7.25682 | 27.50016 | 35.19918 | 31.7952 | 65.17214 | 45.71861 | 64.74509 |
| ظ | 7.2629 | 27.5619 | 35.66102 | 31.49789 | 64.5861 | 45.59582 | 65.9734 |
| ظ | 7.252297 | 26.46113 | 35.62061 | 31.13576 | 64.21236 | 45.02932 | 63.73586 |
| ظ | 7.258091 | 28.5942 | 36.66988 | 32.28203 | 66.50969 | 47.44933 | 67.9143 |
| ظ | 7.247634 | 26.06223 | 35.51523 | 31.17846 | 64.53652 | 47.78491 | 66.53111 |
| ظ | 7.25371 | 27.81337 | 36.485 | 31.43496 | 64.97742 | 47.98685 | 67.97294 |
| ظ | 7.257079 | 27.00771 | 36.08881 | 30.89746 | 63.93454 | 45.61988 | 64.61919 |
| ظ | 7.254752 | 28.22059 | 36.55219 | 31.03288 | 64.33383 | 47.22195 | 67.30556 |
| ظ | 7.257463 | 28.51423 | 37.33693 | 31.15544 | 65.00854 | 46.4818 | 65.40567 |
| ظ | 7.258035 | 27.13425 | 37.20255 | 30.90323 | 64.46539 | 56.76631 | 65.93105 |
| ظ | 7.257004 | 28.04768 | 36.20539 | 31.73106 | 65.74561 | 45.86942 | 64.34023 |
| ظ | 7.262014 | 27.61141 | 37.33374 | 31.15336 | 64.98062 | 47.24891 | 66.16106 |
| ظ | 7.261252 | 27.74731 | 36.58651 | 31.67018 | 66.49078 | 46.09177 | 63.6763 |
| ظ | 7.262156 | 26.84078 | 37.60883 | 31.16949 | 65.46373 | 44.61186 | 63.23587 |

Table (6) Features of testing character ژ

| Character | Mom1 | Mom2 | Mom3 | Mom4 | Mom5 | Mom6 | Mom7 |
|---|---|---|---|---|---|---|---|
| ژ | **7.297592** | **28.42428** | **38.90013** | **31.93332** | **66.90378** | **48.026** | **67.53091** |
| ژ | **7.299254** | **27.94322** | **38.36726** | **31.98547** | **66.67634** | **48.38843** | **66.6846** |
| ژ | **7.29923** | **27.44556** | **38.95245** | **32.31585** | **67.59946** | **46.54681** | **68.72701** |
| ژ | **7.292525** | **27.01533** | **38.216** | **32.07939** | **66.72538** | **46.18154** | **66.74366** |
| ژ | **7.293197** | **27.14121** | **37.7126** | **31.89219** | **66.21634** | **46.55931** | **66.14409** |
| ژ | **7.295342** | **27.20375** | **38.20052** | **31.95046** | **66.54045** | **46.23301** | **66.87901** |
| ژ | **7.297246** | **27.43663** | **38.14999** | **32.14006** | **66.81053** | **46.26984** | **67.49023** |
| ژ | **7.295318** | **28.16774** | **38.88774** | **31.91039** | **66.83506** | **47.4632** | **66.84924** |
| ژ | **7.293839** | **27.04702** | **37.85551** | **31.85924** | **66.24368** | **47.78734** | **66.24185** |
| ژ | **7.295018** | **26.83773** | **37.67548** | **31.81542** | **66.07449** | **45.7641** | **66.28751** |
| ژ | **7.294901** | **27.29787** | **39.24198** | **31.72357** | **66.72309** | **47.70553** | **66.38821** |
| ژ | **7.298668** | **28.28154** | **39.2392** | **32.98768** | **68.61213** | **51.30988** | **68.60284** |
| ژ | **7.295225** | **29.11475** | **39.44118** | **31.64035** | **66.73547** | **47.99948** | **66.27489** |
| ژ | **7.297196** | **28.95444** | **42.01264** | **32.25231** | **69.17109** | **48.20018** | **67.64083** |
| ژ | **7.293068** | **27.44854** | **39.00585** | **31.57003** | **66.41357** | **46.40492** | **65.74163** |
| ژ | **7.286625** | **26.91091** | **36.52428** | **30.871** | **64.10007** | **46.86377** | **64.57683** |
| ژ | **7.291129** | **26.77784** | **37.52318** | **31.69337** | **65.80164** | **46.73041** | **65.73744** |
| ژ | **7.294202** | **27.51459** | **38.34658** | **31.97455** | **66.78398** | **47.64189** | **67.68858** |
| ژ | **7.294542** | **27.81059** | **40.33891** | **32.50827** | **68.61346** | **46.88655** | **66.80622** |
| ژ | **7.292812** | **27.08202** | **38.3423** | **31.73479** | **66.32737** | **47.20064** | **66.12659** |

Table (7) Recognition results of character (ف) written by different writers.

| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ف |
| 2 |  | ض |
| 3 |  | ف |
| 4 |  | ف |
| 5 |  | ف |
| 6 |  | ف |
| 7 |  | ف |
| 8 |  | ف |
| 9 |  | ض |
| 10 |  | ض |
| 11 |  | ف |
| 12 |  | ث |
| 13 |  | ف |
| 14 |  | ف |
| 15 |  | ف |

Table (8) Recognition results of character (ج) written by different writers.

| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ج |
| 2 |  | ج |
| 3 |  | ي |
| 4 |  | ي |
| 5 |  | ي |
| 6 |  | ج |
| 7 |  | ج |
| 8 |  | ج |
| 9 |  | ج |
| 10 |  | ج |
| 11 |  | ج |
| 12 |  | ج |

| | | |
|---|---|---|
| **13** |  | **ج** |
| **14** |  | **ج** |
| **15** |  | **ب** |

Table (9) Recognition results of character (غ) written by different writers.

| **Writers** | **Input character** | **Rec. Results** |
|---|---|---|
| **1** |  | **ص** |
| **2** |  | **غ** |
| **3** |  | **ص** |
| **4** |  | **ص** |
| **5** |  | **ص** |
| **6** |  | **ص** |
| **7** |  | **ض** |
| **8** |  | **غ** |

| | | |
|---|---|---|
| **9** |  | غ |
| **10** |  | ص |
| **11** |  | ص |
| **12** |  | غ |
| **13** |  | غ |
| **14** |  | ص |
| **15** |  | ص |

Table (10) Recognition results of character (ض) written by different writers.

| **Writers** | **Input character** | **Rec. Results** |
|---|---|---|
| **1** |  | ص |
| **2** |  | ص |
| **3** |  | ض |
| **4** |  | ض |
| **5** |  | ض |

| | | |
|---|---|---|
| 6 |  | ض |
| 7 |  | ض |
| 8 |  | ض |
| 9 |  | ض |
| 10 |  | ض |
| 11 |  | ض |
| 12 |  | ض |
| 13 |  | ض |
| 14 |  | ض |
| 15 |  | ض |

Table (11) Recognition results of character (ظ) written by different writers.

| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ض |
| 2 |  | ف |

| | | |
|---|---|---|
| 3 |  | ف |
| 4 |  | ت |
| 5 |  | ت |
| 6 |  | ت |
| 7 |  | ل |
| 8 |  | ل |
| 9 |  | ت |
| 10 |  | ت |
| 11 |  | ط |
| 12 |  | ط |
| 13 |  | ظ |
| 14 |  | ط |
| 15 |  | ظ |

Table (12) Recognition results of character (ز) written by different writers.

| Writers | Input character | Rec. Results |
|---|---|---|
| 1 |  | ز |
| 2 |  | ز |
| 3 |  | ز |
| 4 |  | ل |
| 5 |  | ل |
| 6 |  | ق |
| 7 |  | ذ |
| 8 |  | ك |
| 9 |  | ن |
| 10 |  | ق |
| 11 |  | ط |
| 12 |  | ل |
| 13 |  | ز |

| | | |
|---|---|---|
| **14** |  | **ذ** |
| **15** |  | **ز** |

# الخلاصة


تمييز النمط جزء ضروري في الانظمة المتقدمة لمعالجة الصور.

اللغة العربية تمتلك اربع اشكال لكل حرف بالاعتماد على موقع الحرف من الكلمة. وهذه المواقع هي في بداية الكلمة، الوسط، النهاية او منفصل. هذا البحث يهتم بتمييز الحروف العربية المفصولة باستخدام مفهوم العزوم السبعة.

العزوم ودوال العزوم تستخدم على نطاق واسع كخصائص عامة ثابتة للصور في تمييز الحروف.

تمييز الحرف يتم بمرحلتين الاولى مرحلة التدريب والثانية مرحلة الاختبار وهذا يتضمن العديد من الوظائف الرئيسية تبدا من ادخال الحرف و تنتهي بتمييزه.

هذه الوظائف هي: المعالجة المسبقة، استخلاص الخصائص و مطابقة الحروف. تتضمن المعالجة المسبقة: ادخال الصورة، ازالة الشوائب باستخدام المرشح المتوسط، تحويل الصورة الى صورة ثنائية و اخيرا فصل الحروف.

مطابقة الحروف تتضمن حساب (مسافة اقليدس) بين الحروف المراد اختبارها والحروف المخزونة.

تم تطبيق نظام التمييز على الحروف المطبوعة وكانت نسبة الدقة لتمييز 65 حرف مطبوع 96.92307 بينما في الحروف المكتوبة يدويا نسبة التمييز، قلت نسبة الى عدم الانتظامية التي تظهر في الحروف المكتوبة يدويا.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

# تمييز الحرف العربي بالاعتماد على طريقة العزوم

رسالة

مقدمة الى قسم علوم الحاسبات في جامعة النهرين كجزء من متطلبات
نيل درجة الماجستير في علوم الحاسبات

من قبل

رغد خريبط راشد الخالدي
(بكالوريوس جامعة النهرين 2005)

إشراف

د. ليث عبد العزيز العاني		د.طه سعدون باشاغا

1429		2008