AN OCR SYSTEM FOR ARABIC HANDWRITING

BY

Lowai Kamal El-Deen Abd-Allah

A report submitted in partial fulfillment of the requirements for the degree of:
B.Sc. (Hon.)
In
Electrical and Electronic Engineering

Supervisor:
Mohammed Gaafar El-Nawrany

Department of Electrical and Electronic Engineering
Faculty of Engineering
University of Khartoum

July 2011
DECLARATION OF ORIGINALITY

I declare that this thesis entitled “An Optical Character Recognition (OCR) for Arabic Handwriting” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award except as specified.

Signature: _________________________

Name: ___________________________

Date: ____________________________
DEDICATION

This thesis is dedicated to all family, specially my precious mother and my father-who have supported me with their encouragement, directed me with their wisdom, understood me with their feelings and protected me with their love.
ACKNOWLEDGMENT

I would like to articulate my praise and gratitude to Almighty Allah for all physical and mental support throughout my life and during this project phases.

I would like to express my sincere gratitude to my supervisor, Mohammed Al-Nawrany who was extremely patient, friendly, supportive, helpful and encouraging. I am so thankful to my project partner and my bro Abd-El-Rahman El-Tegany, for his patient, great thinking, the feeling of the group working and his hard works to reach the end of this project. Without his partnership, this project could not have been accomplished.

Finally, I would like to thank my great family, my mother, father, brothers, sisters and the rest for their unlimited support during my life and during the phases of this project.
ABSTRACT

There is strong needing to convert the handwritten characters into the corresponding characters in the computer system; that to be easy and simple in processing (ex: searching, copy and paste, scaling, reshaping …etc) which needed in many applications.

The major objective of this project is to design and implement OCR system for Arabic characters-basic numbers and few letters, and which OCR stands to Optical means that the input to this system is coming as image contains the letter wanted to recognized, Character which here is Arabic letters and numbers, Recognition of the handwritten letter and converts it to computer character, which is needing to study Arabic characters features, segmentation and then designing suitable system for recognition by noticing that- the Arabic language has different characteristics over all the other languages, for that it need different processing, and also using some characteristics that the Arabic letters have in its structure like dots and circles. In this project the concepts of the digital image processing (DIP) are used to find suitable processing for Arabic characters. Also the technology of the artificial neural network (ANN) was used for numbers recognition to determine the corresponding number to the number in the image. Hopfield algorithm was applied for detecting letters. All the phases of this project are done in the Matlab simulator.

The project objective has been successfully accomplished after designing the OCR system for Arabic handwriting. All the components of the design was designed by Matlab simulator, and also has been tested by doing a survey for 53 random people with their own handwriting, and then the data were collected, applied to the system and the results were shown. The result is that- the system is successfully in recognizing the 10 basic Arabic numbers (by certain good performances) and not excellent for letters recognition by using the Hopfield Algorithm.
المستخلص:

هناك حوجة شديدة لتحويل الأشكال المكتوبة بخط اليد إلى الأشكال المقابلة في نظام الحاسوب وذلك نسبة لأن معظم التدشاقات تكون مكتوبة بخط اليد. في حين أن التعامل مع النصوص يكون أسهل (مثل البحث، نسخ وقص النصوص، تغيير الحجم، الشكل...) لذا يمكن أن يدخل في كثير من التطبيقات.

الهدف الرئيسي لهذا المشروع هو تصميم وتنفيذ نظام التعرف الضوئي على الأشكال (OCR) للأرقام و بعض الحروف العربية وهذا الاختصار يعني أن في هذا النظام الدخل يكون في شكل صورة تحتوي على الشكل المراد التعرف عليه وهذا النظام بيد بالأشكال العربية. والتعرف على التعرف على الكتابة بخط اليد وتحويلها إلى حروف الحاسوب. هذا النظام يتطلب دراسة خصائص الأشكال العربية، وضعها في مجموعات ومن ثم تصميم نظام مناسب لتمييز هذه الأشكال العربية. مع ملاحظة أن اللغة العربية لها خصائص مختلفة عن بقية اللغات الأخرى. ولذا تتطلب نوعاً من المعالجة المختلفة عن بقية اللغات وأيضاً يمكن استخدام بعض الخصائص التي تمثلكا اللغة العربية في عملية التعرف على الأشكال مثل النقطة والدوائر.

في هذا المشروع تم استخدام مفاهيم المعالجة الرقمية للصور (DIP) لإيجاد المعالجة المناسبة للأشكال العربية. وأيضاً تم استخدام تقنية الشبكات العصبية الاصطناعية (ANN) في التعرف على الأرقام العربية لتجميع الشكل المقابل للأصل الموجود في الصورة. كما تم تطبيق خوارزمية هوبفيلد للتعرف على الحروف العربية. كل مراحل هذا المشروع تم تنفيذها في برنامج المحاكي MATLAB.

هدف المشروع تم تحقيقه بنجاح بتصميم نظام التعرف الضوئي للأرقام العربية وبنسبة غير كبيرة للحروف العربية. وتم تصميم كل الأجزاء للنظام عن طريق برنامج MATLAB. وأيضاً اختيار بفضل الصحي ل35 شخص عشوائيين بكتابةهم اليدوية الخاصة، وبعد ذلك جمع البيانات وأدخلت النظام، تم التحليل، ومن ثم تم استخلاص النتائج. و النتيجة بأن النظام يعترف بنجاح على العشرة أعداد العربية الأساسية (بعد الأداء الممتاز) وليس بإستثناء التعرف على الحروف باستخدام خوارزمية هوبفيلد.
## List of Abbreviations

<table>
<thead>
<tr>
<th>The Abbreviation</th>
<th>The Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>ACR</td>
<td>Arabic Character Recognition</td>
</tr>
<tr>
<td>Ca</td>
<td>Number of All Connected Components</td>
</tr>
<tr>
<td>DIP</td>
<td>Digital image processing</td>
</tr>
<tr>
<td>DSP</td>
<td>digital signal processing</td>
</tr>
<tr>
<td>H</td>
<td>Number of Holes</td>
</tr>
<tr>
<td>KDD</td>
<td>Knowledge discovery in databases</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language processing</td>
</tr>
<tr>
<td>NN</td>
<td>neural network</td>
</tr>
<tr>
<td>OCR</td>
<td>Optical Character Recognition</td>
</tr>
<tr>
<td>W</td>
<td>The width</td>
</tr>
<tr>
<td>ART</td>
<td>Adaptive Resonance Theory</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure (2.1) RGB color system ..........................................................7
Figure (2.2) Image colors ........................................................................7
Figure (2.3) a colored digital image ..........................................................8
Figure (2.4) components of the digital image in figure (2.3) ....................8
Figure (2.5) Processes of DIP .................................................................9
Figure (2.6) Number ONE representation after thresholding ..................10
Figure (2.7) A 2-dimensional image and its histogram ............................10
Figure (2.8) Artificial Neuron Model .......................................................13
Figure (2.9) Layers of neurons .................................................................14
Figure (2.10) some of the activation functions used in ANN ....................14
Figure (2.11) Block diagram for the supervised learning process .............14
Figure (2.12) methodology for classification ..........................................17
Figure (2.13) Associative network ............................................................18
Figure (2.14) Hopfield network ...............................................................19
  Figure (3.1) flow chart for Hopfield algorithm ....................................21
Figure (3.2) ANN methodology ..............................................................23
Figure (3.3) Nested IF methodology ......................................................23
Figure (3.4) Thresholding Methodology ................................................24
Figure (3.5) descriptor extraction ..........................................................25
Figure (3.6) Numbers recognition using nested IF ..................................27
Figure (3.7) linear and perceptron Neuron Models used in ANN module ....28
Figure (4.1) success percentages for numbers using nested IF and NN …….31

Figure (4.2) the failure percentages for numbers using Neural Network …….32

Figure (4.3) success percentages for selected letters using Hopfield algorithm: 32

Figure (4.4) Performance for numbers in class (I) from the Neural Network …33

Figure (4.5) Performance for numbers in class (II) from the Neural Network ..34

Figure (4.6) Performance for numbers in class (III) from the Neural Network.35

Figure (4.7) Performance for numbers in class (IV) from the Neural Network...36

Figure (4.8) Samples for Undesired dots ..........................................................37

Figure (4.10) Samples for thin bold .................................................................37

Figure (4.11) Samples for Deviation is slope .................................................38

Figure (4.12) Samples for imperfect writing ....................................................38
# TABLE OF CONTENTS:

DECLARATION OF ORIGINALITY .......................................................... i

DEDICATION .................................................................................. ii

ACKNOWLEDGEMENT ...................................................................... iii

ABSTRACT ...................................................................................... iv

المستخلص ...................................................................................... v

LIST OF APPREVIATIONS ............................................................... vi

TABLE OF CONTENTS ................................................................. vii

LIST OF FIGURES ........................................................................... x

LIST OF TABLES ............................................................................. xii

CHAPTER 1
INTRODUCTION .............................................................................. 1

1.1 Overview ..................................................................................... 1

1.2 Problem Definition ................................................................. 2

1.3 Objective .................................................................................... 2

1.4 Methodology, Tools and Knowledge Used ............................... 3

1.5 Thesis Layout ........................................................................... 3

CHAPTER 2

2.1 Previous work ........................................................................... 4

2.2 LITERATURE REVIEW ............................................................. 6

2.2.1 Digital Image Processing ..................................................... 6
2.2.1.1 Basic Concepts & Terminology in DIP.........................................................8
2.2.1.2 DIP Processes..................................................................................................9
2.2.1.2.1 Image Segmentation.....................................................................................9
2.2.1.2.2 Feature(s) Extraction..................................................................................12
2.2.1.2.3 Pattern Recognition....................................................................................12
2.2.2 Artificial Neural Networks..................................................................................13
2.2.2.1 Basic Components & Structure....................................................................13
2.2.2.2 Learning & Training ANN.............................................................................15
2.2.2.3 Real-life applications....................................................................................16
2.2. 3. Pattern Recognition.........................................................................................16
       2.2.3.1 Associative Networks................................................................................18
       2.2.3.2 The Hopfield Model................................................................................19

CHAPTER 3

SYSTEM DESIGN........................................................................................................20

Methodology.............................................................................................................20

3.1 letters..................................................................................................................20

       3.1.1 Hopfield algorithm....................................................................................20
       3.1.1.1 Firstly: calculation of the Hopfield matrix..............................................20
       3.1.1.2 Secondly: recognition of any input letter..............................................21

3.2 Numbers..............................................................................................................22
3.2.1 Thresholding ................................................................. 23
3.2.2 Features extracting ....................................................... 24
3.2.3 Description module ....................................................... 24
3.2.4 Recognition (using Nested IF) module ............................. 26
3.2.5 Recognition (using ANN) module ................................. 28

CHAPTER 4
IMPLEMENTATION & RESULTS ........................................... 30
4.1 Implementation .............................................................. 30
4.2 Results ........................................................................... 31
4.3 Analysis .......................................................................... 37
   4.3.1 Misrecognition or misclassification causes ..................... 37
   4.3.2: Collected Data .......................................................... 39

CHAPTER 5
CONCLUSION & FUTURE WORKS ..................................... 40
5.1 Conclusion of the Project ............................................... 40
5.2 Future Works ................................................................. 42
REFERENCES ......................................................................... 43
APPENDICES
Appendix A ........................................................................... A1
Appendix B ........................................................................... B1
LIST OF TABLES

Table (3.1) designing NN for class (I) ..............................................28
Table (3.2) designing NN for class (II) ............................................28
Table (3.3) designing NN for class (III) .................................29
Table (3.4) designing NN for class (IV) ............................................29
Table (4.1) shows the results for numbers recognition ...........31
Table (5.1) performance of the recognition ..............................41
CHAPTER ONE

Introduction

1.1 Overview:

Digital image processing has become a vast domain of modern signal technologies. Its applications pass far beyond simple aesthetical considerations, and they include medical imagery, television and multimedia signals, security, portable digital devices, video compression, and even digital movies. We have been flying over some elementary notions in image processing but there is yet a lot more to explore.

Pattern Classification, more often called Pattern Recognition, is the primary bottleneck in the task of automation. Pattern recognition is the research area that studies the operation and design of systems that recognize patterns in data. It encloses sub-disciplines like discriminate analysis, feature extraction, error estimation, cluster analysis, grammatical inference and parsing. Important application areas are image analysis, character recognition, speech analysis, man and machine diagnostics, person identification and industrial inspection.

Language is a system of signs (indices, icons, symbols) for encoding and decoding information. Language is the most important channel of communication between peoples of different cultures and different countries.

In the past, all most contracts, letters, recommendations, billing….etc were done and handled by the hand writing. Now a days, softcopies are the most popular in handling due to its easiest in copy, cut-paste, transfer, less size, save time, reduce cost, good presentations…etc.

For this importance of the language in human life, language processing takes space in the computer sciences of pattern recognition by field named Natural Language processing (NLP) which is a field of computer science and linguistics (language sciences) concerned with the interactions between computers and human (natural) languages. There are two branches in NLP first is the natural language generation systems which convert information from computer databases into
readable human language, the second branch is the **natural language understanding** systems which convert samples of human language into more formal representations such as parse trees or first-order logic structures that are easier for computer programs to manipulate\(^1\)[3][9].

A basic goal of NLP is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs, and that by making the computer generates and understands the natural languages.

### 1.2 Problem Definition:

To computerize an old record-keeping system in an office, to do this by typing by the hand it spends more time and effort and it is tedious task for the human, some programs can be constructed under the natural language processing field to do this process easily.

Sometimes there are documents written by hand and it is wanted to input these documents to the computer to be manipulated by the computer (manipulating by computer so easy rather than the human, manipulating like searching, copy and paste,...etc, it becomes very easy using the computer) if it done by typing, it takes more times and efforts, and also sometimes the handwriting not easy to make recognition to the letter.

Some applications need to make recognition for the letters which written by hand before manipulating the text in some way, as examples: electronic exam correction and for application of the dictionary in translation handwritten text from one language to other language.

As a useful application, for the blind man to listen to the handwritten text, firstly the handwritten text must be converted to computer text, after that using the speech recognition to convert the text to sound to help the blind man.

### 1.3 Objectives:

Firstly: To design and develop an algorithm to detect & recognize Arabic numbers handwritten and introduces the corresponding character in the computer system.

Secondly: to design and implement the Hopfield algorithm to recognize Arabic handwriting letters.
1.4 Methodology, Tools and Knowledge Used:

MATLAB simulator will be used to manipulate the image of the character. The design and development processes will take place in MATLAB environment.

In numbers recognition, the descriptors gathering will be out manually, choose the best design according to these descriptors by Neural Networks and another one using normal nested If- as a reference of compression, pass the desired image to be recognized to the program and then get the results.

For letters recognition, Hopfield algorithm is used to detect the selected letters and the final results are analyzed and represented in graphs.

Knowledge used: Digital Image Processing (DIP), Artificial Neural Network (ANN).

First in this project make overview about the DIP and how to use it in this project, after that, discuss the Arabic letters features and choose suitable descriptors, then discuss the neural network (NN) and how to use it in this project. After design complete, making testing for this project by data collected randomly.

1.5 Thesis Layout:

This thesis contains five chapters, this is chapter one and the rest is as following:

Chapter two: literature view: this chapter discusses the previous work in the OCR system for Arabic characters, basic concepts, Hopfield algorithm and what is new in this project.

Chapter three: design of the OCR system for Arabic numbers handwriting according to its descriptors, define the components of the system and implement the Hopfield algorithm for selected letters.

Chapter four: results and implementation: of the OCR systems and discuss the results.

Chapter five: conclusions and future works: conclusions of this project and recommendations for the future works.

Appendix A: code of Matlab.

Appendix B: form of questionnaire for collecting data and some samples.
CHAPTER TWO

Project Theory

2.1 Previous Work

Working in pattern recognition has an old history, starting from the 18's. First tries about classification was based on special primary electrical circuits by entering a manual input.

As human increase his thinking level, his desire become hungrier- so when the images became popular for people, it must be in functional form to handle it in easy and simple form.

There are several algorithms to handle this based on some features: [11]

Classification algorithms (supervised algorithms predicting categorical labels)

- Maximum entropy classifier (aka logistic regression, multinomial logistic
  Decision trees, decision lists
- Support vector machines
- Kernel estimation and K-nearest-neighbor algorithms
- Perceptrons
- Neural networks (multi-level perceptrons)

Clustering algorithms (unsupervised algorithms predicting categorical labels)

- Categorical mixture models
- K-means clustering
- Hierarchical clustering (agglomerative or divisive)
- Kernel principal component analysis (Kernel PCA)

Regression algorithms (predicting real-valued labels)

Supervised:

- Linear regression and extensions
- Neural networks
- Gaussian process regression (kriging)
Unsupervised:

- Principal components analysis (PCA)
- Independent component analysis (ICA)

**Categorical sequence labeling algorithms** (predicting sequences of categorical labels)

Supervised:

- Hidden Markov models (HMMs)
- Maximum entropy Markov models (MEMMs)
- Conditional random fields (CRFs)

Unsupervised:

- Hidden Markov models (HMMs)

There many applications for this classification such as: image processing, handwriting recognition, face or finger recognition… etc.

Anyway, even more interesting than the history lesson is the discussion of different types of character recognition, starting with plain old OCR- which was using the inputs in its functional form and compared it to a references stored in databases using XY position- and going all the way up to NHR (Natural Hand Recognition).

Few works have been done on Arabic character & handwriting, due to its complexity.

The methods used for recognition, can be divided into 2 categories: segmentation-free methods and segmentation-based methods. Gillies used the approach of over segmenting the word to insure that no segment belongs to more than one character, and then he used the combined features as classifiers and passed them to a neural network. Elgammal & Ismail segmented the word to scripts (small connected segments) using a Line Adjacency Graph (LAG) then they used the features of these scripts as classifiers.
In segmentation-free methods, a wide range of different features have been used by different groups. Many of these features require normalization of the image before features can be extracted. An important part of this is often finding the baseline of the character (the line on which it is written and by which the characters are often connected). Other techniques have been developed in this area such as: Nearest Neighbor method which was used on a small dataset containing 20 characters, but it required resizing of the image. Peter Burrow described a method called radial method used to obtain statistical features of the characters. This method had very poor performance.

Ameen M. El-Ameen and Ahmed Abu Algasim, who were got degree of B.Sc., based on neural network to recognize Arabian characters, but they got weak performance and no-good recognition in their experiment.

In this thesis, it is intended to perform Arabic character recognition using few topological descriptors and a proposed descriptor using neural networks and MATLAB program.

2.2 Literature Review:

Here comes a brief description of the literature related to the work in the area of the Arabic Character Recognition (ACR). Recognition basically depends on Digital Image Processing for the manipulation of the image and extracting features and on either programming the recognition algorithms or Artificial Neural Networks (ANN) for the recognition. This means there are three main topics:

1- Digital image processing.

2- Artificial neural networks.

3- Recognition algorithms.

2.2.1 Digital Image Processing

Digital image processing (DIP) is a science in which a special kind of signals known as images can be manipulated within a computer system in order to obtain and extract some information that has to do something with the objective(s) of the manipulation. It is a discipline of DSP (digital signal processing). [4]
A digital image is represented as a 2-dimensional matrix in the computer system, by ignoring the third one-depth- in images. It is a function of 2 variables \( f(x,y) \) where \( x \) and \( y \) indicate the row and the column in which the pixel lies horizontally and vertically respectively. The value of the function \( f(x,y) \) represents the gray level associated with the pixel \((x,y)\) and it varies from 0 (black) to 255 (white) and the midway is a mixture of white and black (and here comes the name gray-levels).\(^5\)

The above description applies for the grayscale image. The colored image is represented as a 3-dimensional matrix where the 3\(^{rd}\) dimension indicates the RGB (Red, Green, and Blue) components.

Every image can be represented as two-dimensional array, where every element of that array contains color information for one pixel.

![Image Colors](image Colors.png)

**Figure (2.1) RGB color system**

Each color can be represented as a combination of three basic color components: red, green and blue.
A colored image can be visualized as 3 images placed as layers; each image is a grayscale image with different variations from each other.

Figure (2.3) a colored digital image

Figure (2.4) (From left to right) red, green, and blue components of image in 2.3

2.2.1.1 Basic Concepts & Terminology in DIP

- **Connectivity**: two pixels are said to be connected if they are neighbors and their gray levels satisfy a specified criterion of similarity (say, if they are equal).

- **Region**: let \( R \) be a subset of pixels in an image, \( R \) is a region if \( R \) is a connected set.

- **Boundary (Contour or Border)**: a boundary of a region \( R \) is a set of pixels in the region that have one or more neighbors that are not in \( R \).
2.2.1.2 DIP Processes:

There are 3 basic processes in the digital image processing:

i. Image Segmentation.

ii. Feature(s) Extraction.

iii. Pattern Recognition.

![Diagram showing the processes of DIP](image)

**Figure (2.5) Processes of DIP**

2.2.1.2.1 Image Segmentation

Segmentation is the process in which the image is subdivided into object(s) & background. It is the process by which the computer system knows which pixels belong to the object that is required to be manipulated and which ones must be discarded as they belong to the background.

There are many techniques used to segment images such as: detection of discontinuities, Hough Transforms, and thresholding.

Thresholding is the mapping of the values of the pixels of a 2-dimensional image into one of 2 values (usually 0 or 1) using a specified threshold. It is the process of converting a 2-dimensional image into a binary image. A binary image is that whose pixels have only one of 2 values; these values are labels used to refer to either object or background. The threshold is used to divide the image into 2 regions: region with gray level values greater than the threshold and other with gray level values less than the threshold.

A thresholded image g(x,y) for an image f(x,y) with threshold (T) is defined as follows:

\[
g(x, y) = \begin{cases} 
1 & , \quad f(x, y) > T \\
0 & , \quad f(x, y) \leq T 
\end{cases}
\]  

(2.1)
As an example, after thresholding, number ONE can be extracted as an array as follow:

![Array representation of number ONE after thresholding](image)

**Figure (2.6) Number ONE representation after thresholding**

The thresholding process is histogram-based; it requires knowing the histogram of the image to define the threshold that can divide the image into its object(s) and background. A histogram of a digital image with gray levels in the range \([0, L-1]\) is the discrete function:

\[
h(r_k) = n_k
\]  

(2.2)

Where:

- \(r_k\): the \(k\)th gray level.
- \(n_k\): the number of pixels in the image having gray level \((r_k)\).
Figure (2.7) A 2-dimensional image (left) and its histogram (right)

The threshold value is computed using several mathematical methods. One of these methods is to compute the threshold from the image’s statistics (the histogram) using Otsu’s method which is described as follows\[^3\]:

- Evaluate the normalized histogram for the image, i.e., treat the histogram as a discrete probability density function as in:
  \[
  p_r(r_q) = \frac{n_q}{n} \tag{2.3}
  \]

- Suppose that a threshold \((k)\) is chosen such that \(C_0\) is the set of pixels with levels \([0,1,2,\ldots,k-1]\) & \(C_1\) is the set of pixels with levels \([k,k+1,k+2,\ldots,L-1]\).

- Choose the value of \((k)\) that maximizes the between-class variance \(\sigma_B^2\).
  \[
  \sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 \tag{2.4}
  \]

Where:

\[
\omega_0 = \sum_{q=0}^{k-1} q p_r(r_q) \tag{2.5}
\]

\[
\omega_1 = \sum_{q=k}^{L-1} q p_r(r_q) \tag{2.6}
\]

\[
\mu_0 = \sum_{q=0}^{k-1} q p_r(r_q) / \omega_0 \tag{2.7}
\]

\[
\mu_1 = \sum_{q=k}^{L-1} q p_r(r_q) / \omega_1 \tag{2.8}
\]

\[
\mu_T = \sum_{q=0}^{L-1} q p_r(r_q) \tag{2.9}
\]

- The value of \((k)\) is the threshold and it is a normalized value from 0.0 to 1.0.

This threshold is done in MATLAB by:

```matlab
x = graythresh(y);
```

This gives the value of \((k)\) above.
2.2.1.2.2 Feature(s) Extraction

After an image has been segmented, the resulting binary image is represented and described in a form suitable for further computer processing. The task is divided into 2 tasks: representation and description. Representation is done by selecting one of the representation schemes used such as: chain codes. Description is done after representation is finished. It is a process in which the boundary or the region is described.

There are 2 types of descriptors:

- Boundary descriptors such as: length, diameter, curvature, eccentricity…etc.
- Regional descriptors which are divided into 2 categories: simple & topological.

Simple regional descriptors include area, perimeter, compactness, mean, and median of the gray level.

Topological descriptors are useful for global description of regions in the image plane. Topology is the study of properties of a figure that are unaffected by any deformation, as long as there is no tearing or joining of the figure. Topological descriptors include the following:

i. Number of Holes (H).
ii. Number of All Connected Components (Ca).
iii. The width (W).

This done in MATLAB using:

\[ [L, c] = bwmorph(g); \]
\[ D = regionprops(L); \]

2.2.1.2.3 Pattern Recognition

After an image’s descriptors are obtained they are arranged as patterns (or objects) for the recognition phase. A pattern is defined as an arrangement of descriptors (features). A pattern class is a family of patterns that share some common properties. Patterns are denoted symbolically as vectors.

Many techniques are used for pattern recognition. One of the famous techniques is the artificial neural networks which will be described briefly in the next section, another method is to classify each character alone as a pattern then they can be recognized by a simple code.
2.2.2 Artificial Neural Networks

An artificial neural network may be defined as a network of units called neurons (the basic processing elements for the network) which communicate by sending signals to each other over a large number of weighted connections.

It is known that the human brain is estimated to have around 10 billion neurons each connected on average to 10000 other neurons forming a complicated network of neurons. The artificial neural networks were inspired by biological findings relating to the behavior of the brain.

2.2.2.1 Basic Components & Structure

The basic components of an artificial neuron may be summarized as follows:

- Inputs ($x_i$).
- Connecting weights ($w_i$).
- Effective input (or net input) ($v$).
- Output ($y$).
- Activation function ($f(v)$) which maps the input to output.

\[
\begin{align*}
    v &= w_0 + \sum_{i=1}^{n} w_i x_i \\
    y &= f(v)
\end{align*}
\]

Figure (2.8) Artificial Neuron Model
The neurons are usually arranged in layers to form a neural network. Each layer of neurons receives outputs from the previous layer as inputs, calculates its outputs, and delivers them to the next layer.

![Figure 2.9 Layers of neurons](image)

There is several activation functions used in the neural networks. Some of them are shown in figure 2.10 below. The choice of the activation function is based upon the purpose of the neural network design. The chosen function must be able to classify given input pattern vectors into the desired classes (which are called targets).

![Figure 2.10 some of the activation functions used in ANN](image)

Neural networks can be classified into architectures based on the type of the activation function used. One of the famous architectures is the perceptron which is a neuron with a hard limiting activation function.
2.2.2.2 Learning & Training ANN\(^1\)

The most important characteristic of an artificial neural network is that it can learn and detect regularity & irregularity of the input data. There are several learning rules and algorithms developed to perform the learning process; each group of rules is applicable to certain structures of neural networks. The concept of the learning is based upon the modification of the connecting weights so that the mean square error (the mean square of the difference between the desired output & the actual output of the network) is as less as possible (the goal is ideally zero) and the relationship between the input and the output is estimated with more accuracy. This process is similar to the curve fitting problems in numerical analysis where certain points in an n-dimensional space are given and a curve that passes through them is required to be estimated. The learning in which the targets are known is called supervised learning.

![Block diagram for the supervised learning process](image)

**Figure (2.11) Block diagram for the supervised learning process**

Training is the process of collecting as many as possible (input pattern, output) pairs from the application domain where the neural network is intended to be implemented, and presenting them in an appropriate form to it, so that its weights are adjusted according to the learning rule embedded within it to map the inputs to the desired outputs. An important terminology in the training is the epoch which is defined as a one pass through all the training data.
2.2.2.3 Real-life applications:

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Robotics, including directing manipulators, Computer numerical control.

Application areas include system identification and control (vehicle control, process control), quantum chemistry, game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

2.2. 3. Pattern Recognition

For the past several years, pattern recognition has been used in a number of areas such as medicine, weather forecasting, automated industrial inspection, geology and agriculture.

Pattern recognition is concerned with the automatic detection or classification of objects or events. Most pattern recognition tasks are first done by human beings and automated later. Automating the classification of objects using the features as those used by people can be a very difficult task. The features used by machines are not precisely the same as those used by human beings. Features that would be impossible or difficult for humans to estimate are useful in automated systems. For example some systems classify objects in satellite images using the wavelengths of light that are invisible to human beings.
For good classification, the useful methodology can be represented as the follow flow chart:

![Flow Chart]

**Figure (2.12) methodology for classification**

We live in a world where events never repeat exactly. Even though events are never exactly the same, they are also not completely different. There is some sort of continuity, similarity and predictability that allows one to generalize correctly from past experience to future events. Neural networks are based on the idea that one can reproduce the power of the human brain by artificial means. The problems where artificial neural networks are very promising include signal processing, speech recognition, visual perception, control and robotics. [8]

The neural networks help to solve these problems with natural mechanisms of generalization. Suppose one represents an object in a network as a pattern of activation of several units. If a unit or two responds incorrectly, the overall pattern any how remains the same and the neural network responds correctly to the stimuli. When neural networks operate, similar inputs naturally produce similar outputs. Most real-world perceptual problems have this structure of input-output continuity.

Here two of the common algorithms of pattern recognition using ANN:

**2.2.3.1 The Nested IF Model**

Her we develop a program using suitable programming language, this code is developed by the nested if statement where it has to define the properties of each pattern and then conditions for every pattern must be made, when we run the program for a certain input all the conditions will be checked if one of them complied then the corresponding pattern is the result of the recognition.
2.2.3.2 Associative Networks

Assume that we have $m$ number of $n$-dimensional bipolar vectors $x_1, x_2, \ldots, x_m$; where $m < n$ (all elements can be +1 or -1), and these must be stored in a memory (Neural Network). The goal is to present a distorted version of any vector $x_i$, then the network must give out $x_i$ correctly.

Assume $m$ pairs of vectors $(x_1; y_1)(x_2; y_2); \ldots; (x_m; y_m)$ and by presenting a distorted vector $x_i$, the correct $y_i$ must be got. This is called heteroassociation. In case of getting the output from the same input, this is called autoassociation ($x$ is associated with itself).

This can be solved by a single layer network with step-functions ($\text{sign}(\varepsilon)$)

![Associative Network Diagram]

**Figure (2.13) Associative network.**

To determine the weight matrix - the Hebbian postulate was used, which states that the elements of $x$ and $y$ which are correlated should increase the corresponding weights of $W$. $W$ is the correlation matrix between $x$ and $y^{[9]}$.

Given $m$ pairs of vectors $(x_i; y_i), i = 1; 2; \ldots; m$, $W$ can be created as

$$w = \sum_{i=1}^{m} y_i x_i^T = YX^T$$  \hspace{1cm} (2.10)

Where:

$Y = [y_1, y_2, \ldots, y_m]$ and $X = [x_1, x_2, \ldots, x_m]$

With creating an autoassociative memory $W = X^*X^T$ (i.e. $y_i = x_i$).
Another algorithm is one of the recurrent networks

### 2.2.3.3 The Hopfield Model

One way to improve the autoassociation is to build a recurrent network of neurons in one layer with step-functions.

![Hopfield network](image)

*Figure (2.14) Hopfield network.*

Where the $Z'$ is a delay operator and $S_i$ is state $i$.

The definition of the model is:

- all outputs are fed to all input
- there is no self-feedback (i.e. $w_{ii} = 0$) The diagonal of $W$ is zero
- the weight matrix $W$ is symmetric ($W = W^T$)
- the states $s_i$ is updated randomly one at each time

Hebbian learning can be used to find $W$. Since in autoassociation a symmetric matrix was got (autocorrelation function are symmetric) but with one modification, the diagonal must be zero.

The last term in the equation makes the diagonal zero, where $I_{n \times n}$ is the identity matrix.

**Steps:**

1. store the vectors $x_1, x_2, \ldots, x_m$ to calculate $W$.

The vectors $x_i$ have elements of +1 or -1. Once the weights $W$ are calculated they are fixed.

2. Let $x$ be a unknown noisy vector presented to the network initialize the state $s(0) = x$

3. Iterate until convergence. Update the elements of the state vector $s$ one at the time.

Repeat updating until $s$ remains unchanged.

4. The output of the network is the final state vector $s$ (i.e. $y = s$).
CHAPTER THREE

Designing

Methodology

In this chapter, how creating the algorithms was be introduced, how the data was prepared in order to be recognized, how descriptors for the characters were found and the classes of these number according to the characteristics of the characters.

3.1 letters:

Firstly three letters were picked and Hopfield algorithm was used to recognize them, the reason for choosing three letters is that the Arabic letters are much closed to each other in the shape so it is very difficult to recognize them all. Secondly the Hopfield algorithm is relatively weak algorithm for similar inputs so it is works good with small and widely differently data.

The chosen letters are (أ،خ،ه)، they have been chosen according to their deviations from each other so they can be recognized correctly.

3.1.1 Hopfield algorithm:

3.1.1.1 Firstly: calculation of the Hopfield matrix:

As explained in the previous chapter matrix W can be computed as follows:

1- Bringing standard photos for the five letters to be the targets.
2- Changing the photos from three to two dimensional arrays.
3- Saving those photos as binary photos (Make thresholding -just zeros and ones).
4- Reshaping the arrays into column vectors.
5- Replacing each zero by minus one.
6- Multiplying each vector by its transpose then adding the results together.
7- The result is the Hopfield matrix (W).
3.1.1.2 Secondly: recognition of any input letter:

1- Entering the letter as an image.
2- Changing the image into two dimensional arrays.
3- Thresholding the input.
4- Replacing each zero by minus one.
5- Reshaping the array as a column vector.
6- Multiplying the Hopfield matrix by the input vector.
7- If the result equals the input then reshapethe result as a final output.
8- Else replace the input vector by the result vector and step 6 must be repeated.

As shown in next flow chart:

---

Figure (3.1) flow chart for Hopfield algorithm
3.2 Numbers:

Number recognition is achieved by passing input number as an image through three steps:
1- Converting to two dimensional array and thresholding.
2- Extract features.
3- Recognition.

Two methods were developed to recognize the Arabic numbers, one of them is using Artificial Neural Networks (ANN) and the other is by programming using traditional nested if statement as a base of comparison, But both methods are same in the first thresholding the image and extract the characteristics from the descriptors.

Arabic numbers can be classified into four different classes as follows:

**Class I: for (5, 9)**

The five and nine: both have a hole.

**Class II: for (0, 1)**

The zero and one: their width is very small.

**Class III: for (7, 8)**

The seven and eight: if the image was divided into tow images horizontally and then the summation of all connected parts was calculated in the two images, they have three connected parts.

**Class IV: for (2, 3, 4, 6)**

The two, three, four and six: they are the rest of numbers.
ANN method:

3.2.1 Thresholding:

It is intended to threshold the input image using the following steps:

- Convert the input digital image into a 2-dimensional image.
- Compute the normalized threshold value from the 2-D image using Otsu’s method described earlier.
- Multiply the normalized value by 255 to obtain the actual threshold value.
- Threshold the image was used by obtained threshold value.
3.2.2 Features extracting:

In this module, the input number was segmented, manually, in its class according to its prosperities (descriptors).

3.2.3 Description module

After thresholding, the following steps are done to extract the features of the image:

1- Obtain the negative of the image (that is because MATLAB can extract features when the background is darker than the text).

2- Obtain the binary image (label matrix) of the image (that is because the label matrix contains some properties of the image).

3- Obtain the area from the label matrix.

4- Obtain the number of connected parts to each half (upper half and lower half) independently, and then add them together to get the total number of connected parts.

5- Calculate the lower width (the width of the lower half of the image) from the extreme, which found in the label matrix.
6- Calculate the upper width (the width of the upper half of the image) from the extreme, which found in the label matrix.

7- Calculate the overall width of the original image, from the bounding box, which found in the label matrix.

8- Calculate the overall high of the all image, from the bounding box, which found in the label matrix.

9- Calculate the left density (number of points in the left half/ the total points).

10- Calculate the upper density (number of points in the upper half/ the total points).

11- Calculate the upper line density (this density used to distinguish between the two and three because they are typical to each other except the upper line in two is not in three).

After doing the above steps to the picture the image is now very clear to be recognized, just a brain to think what is this number is need?, two methods were developed one is using neural network which is a very clever network that can thinks well and recognize the number correctly, the other method is using nested if in a MATLAB program and give this program rules to follow (if.....then......,if.....then....... , etc) but this way is restricted to the conditions which we made, does not have a brain to think, in the next chapters we will introduce a comparison between the two methods and their results.

![Figure (3.5) descriptor extraction](image)

### 3.2.4 Recognition (using Nested IF) module:
In this module a code using nested if was designed to recognize the desired number and depending on the features calculated from the previous module, the algorithm is as follows:

**Zero:**

Has an area which is very small relative to the whole image, and very small high comparing to the image high.

**One:**

Has a very small ratio between width to high.

**Two:**

Has upper density more than lower and left density more than right and has upper line.

**Three:**

Typical to two except it does not have upper line.

**Four:**

Has left density more than right, but lower density more than upper.

**Five:**

Has one hole but this hole is distributed between upper half and lower half.

**Six:**

Has upper density more than lower, but the left density is smaller than the right.

**Seven:**

Has two connected parts in the upper half image.

**Eight:**

Has two connected parts in the lower half image.

**Nine:**

Has one hole but this hole is in the upper half image.
The program is coded that if it find a certain features in the input image the output is the corresponding number, (as in appendix A for the code).

**Figure (3.6)** Numbers recognition using nested IF
3.2.5 Recognition (using ANN) module:

In this module, a perceptron or a linear neural network (a neural network with perceptron) will be used for each class of numbers except those with more than two numbers; the linear neural network will be used to those. The network will consist of only one neuron with one weight and one bias. The input will be the features. The output will be a scalar quantity represents the number. Both weight and bias values will be determined upon training.

![Neuron Models used in ANN module: linear (left) and perceptron (right)](image)

Table (3.1) designing NN for class (I)

<table>
<thead>
<tr>
<th>Training data</th>
<th>Network Type</th>
<th>Targets</th>
<th>Meaning of targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of holes in the upper half image.</td>
<td>Perceptron</td>
<td>One of the numbers [0 1]</td>
<td>0: 5</td>
</tr>
</tbody>
</table>

Table (3.2) designing NN for class (II)

<table>
<thead>
<tr>
<th>Training data</th>
<th>Network Type</th>
<th>Targets</th>
<th>Meaning of targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of connected parts in the upper half image.</td>
<td>Perceptron</td>
<td>One of the numbers [0 1]</td>
<td>0: 7</td>
</tr>
</tbody>
</table>
Table (3.3) designing NN for class (III)

<table>
<thead>
<tr>
<th>Training data</th>
<th>Network Type</th>
<th>Targets</th>
<th>Meaning of targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ratio of the width over the high of the number.</td>
<td>Perceptron</td>
<td>One of the numbers [0 1]</td>
<td>0: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: 1</td>
</tr>
</tbody>
</table>

Table (3.4) designing NN for class (IV)

<table>
<thead>
<tr>
<th>Training data</th>
<th>Network Type</th>
<th>Targets</th>
<th>Meaning of targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left density</td>
<td>Linear</td>
<td>One of the numbers [0 1 2]</td>
<td>0: 2 or 3</td>
</tr>
<tr>
<td>Upper density</td>
<td></td>
<td></td>
<td>1: 4</td>
</tr>
<tr>
<td>Line density</td>
<td></td>
<td></td>
<td>2: 6</td>
</tr>
</tbody>
</table>
CHAPTER FOUR

Implementation, Analysis & Results

In this chapter, the system implementation is discussed as well as the mechanism by which the system was tested. Testing results are also interpreted, in addition to the error's causes.

4.1 Implementation:

Firstly, the M-files must be loaded "as in appendix A ".

Secondly, the desired picture must be resized to the specific scale.

Thirdly, the scaled picture must be uploaded as an input to the MATLAB program.

Lastly, the program should be run in order to get the output.

A form -as in appendix B - was distributed for a random 53 people inside the faculty demanding them to fill it with their handwriting. A survey was carried out-samples as in appendix B- for all 53 persons; each one wrote his own 10 samples for the number from 0 to 9 in Arabic form and the three selected letters, That to say, there are different 689 samples must be recognized, which is a huge number, so for each number or letter a random 25 samples were been chosen.

Scanners were used for data acquisition of images (as seen in appendix B ). Images of each character were subjected to the system and the results were collected. The following Tables show the results obtained from the testing.
4.2 Results:

Table (4.1) shows the results for numbers recognition

<table>
<thead>
<tr>
<th>Sets</th>
<th>Total Correct using nested IF (%)</th>
<th>Total Correct using Neural Network (%)</th>
<th>Mutual correct (%)</th>
<th>Total Wrong Or Unknown by both (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>80</td>
<td>60</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td>One</td>
<td>24</td>
<td>84</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>Two</td>
<td>52</td>
<td>56</td>
<td>52</td>
<td>44</td>
</tr>
<tr>
<td>Three</td>
<td>76</td>
<td>64</td>
<td>64</td>
<td>24</td>
</tr>
<tr>
<td>Four</td>
<td>76</td>
<td>80</td>
<td>76</td>
<td>20</td>
</tr>
<tr>
<td>Five</td>
<td>96</td>
<td>76</td>
<td>76</td>
<td>4</td>
</tr>
<tr>
<td>Six</td>
<td>96</td>
<td>72</td>
<td>72</td>
<td>4</td>
</tr>
<tr>
<td>Seven</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>8</td>
</tr>
<tr>
<td>Eight</td>
<td>84</td>
<td>68</td>
<td>68</td>
<td>16</td>
</tr>
<tr>
<td>Nine</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure (4.1) success percentages for numbers using nested IF and Neural Network
Figure (4.2) the failure percentages for numbers using Neural Network

Figure (4.3) success percentages for selected letters using Hopfield algorithm
Figure (4.4) Performance for numbers in class (I) from the Neural Network
Figure (4.5) Performance for numbers in class (II) from the Neural Network
Chapter 4

Implementation, Analysis and Results

Figure (4.6) Performance for numbers in class (III) from the Neural Network
Figure (4.7) Performance for numbers in class (IV) from the Neural Network
4.3 Analysis:

For both letters and numbers recognition:

4.3.1 Misrecognition or misclassification, can be due to:

a. Undesired dots caused by mistaken or dirt which looks like:

![Image of undesired dots]

**Figure (4.8) Samples for undesired dots**

b. Thin bold or uncompleted lines (which lead to wrong number of connecting part):

![Image of thin bold]

**Figure (4.10) Samples for thin bold**
c. Deviation is slope:

![Deviation is slope](image1)

**Figure (4.11) Samples for deviation is slope**

d. Wrong classification due to imperfect writing-out of the reference, which leads to confusing between numbers:

![Imperfect writing](image2)

**Figure (4.12) Samples for imperfect writing**
4.3.2: Data from the graphs:

i. Recognition for numbers eight and zero has a good percentage for both ways (using nested IF and Neural Network) because of clarity of its form.

ii. The output result of using nested IF for numbers (one, four, six, seven and nine) overcomes the output of Neural Network; due to the clear condition in nested IF and some nonfunctional form out of Neural Network range.

iii. Number five has –approximately – equal percentage for both ways, because of its simplicity form.

iv. In number two, using nested IF has a quite success; because of unconditional structure, but Neural Network trains itself successfully and gets an excellent result. There is a big benefit of using Neural Network, which it is clear from the graph.

v. Nested IF can't recognize number three unless it is in its functional structure. In Neural Network, there are a little different between number two and number three form, in addition to the complex form for number three. That explains the low percentage for number three output.

vi. Hopfield algorithm has a poor performance for imperfect structure.

vii. There are many ways for writing Arabic letters - at the beginning, the middle and in last, which make the recognition very hard.
4.3.3 Comparison between the algorithms

Two algorithms were used for recognition nested If and neural network algorithm few points stated blew to compare them.

i. The nested if algorithm is built in conditions and restricted to them and can’t cross these conditions while the neural network is the opposite so it is construct to work like the human brain so it can think.

ii. The output result of using nested IF for numbers (one, four, six, seven and nine) overcomes the output of Neural Network; due to the clear condition in nested IF and some nonfunctional form out of Neural Network range.

iii. In number two, using nested IF has a quite success; because of unconditional structure, but Neural Network trains itself successfully and gets an excellent result. There is a big benefit of using Neural Network, which it is clear from the graph.

iv. Nested IF can't recognize number three unless it is in its functional structure. In Neural Network, there are a little different between number two and number three form, in addition to the complex form for number three. That explains the low percentage for number three output.

v. The Hopfield algorithm is the worst algorithm as seen from results and that is because it dependence in matrices multiplication so to give a precise results it requires that the matrices are small in size, so the images of characters are in high resolutions which means their matrices have very large sizes and so huge number of pixels.
CHAPTER FIVE:

Conclusion and Future Works

5.1 Conclusion of the Project

i. Comparing the resulting system with the predefined design goals, it can be stated that, the system has accomplished its proposed objectives partially. This project is done, approximately successfully for numbers and in poorly using Hopfield algorithm.

ii. In this project recognition of basic 10 Arabic numbers is done using the Neural Networks as a main recognizer and nested IF as a reference with performance shown in table (5.1). By the NN based on descriptors for recognizing Arabic numbers, the class which the letter belongs to- is specified, inside the class the ratio is used to differentiate between the members of the same class. The assumption behind the ratio is that: the humans write letters with different widths and different heights but still the ratio between the width and height closely constant.

iii. To make wide testing for this project, data were collected from 53 students of Electrical department, Faculty of Engineering, University of Khartoum, subjected to the system and the results were analyzed and presented.

iv. ANNs have the ability to adapt, learn, generalize, cluster or organize data; that to say, it is flexible and can think. While nested IF algorithm is a successful in functional form only, that to say, it is unthinkable.

v. Hopfield algorithm is weak for imperfect structure, wide changing and big images (big number for pixels).

vi. Creating an algorithm with hundred percentage success is quiet impossible, due to different in fonts, size and imperfect writing.
### Table (5.1) performance of the recognition

<table>
<thead>
<tr>
<th>Sets</th>
<th>Performance of the recognition (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>60</td>
</tr>
<tr>
<td>One</td>
<td>84</td>
</tr>
<tr>
<td>Two</td>
<td>56</td>
</tr>
<tr>
<td>Three</td>
<td>64</td>
</tr>
<tr>
<td>Four</td>
<td>80</td>
</tr>
<tr>
<td>Five</td>
<td>76</td>
</tr>
<tr>
<td>Six</td>
<td>72</td>
</tr>
<tr>
<td>Seven</td>
<td>92</td>
</tr>
<tr>
<td>Eight</td>
<td>68</td>
</tr>
<tr>
<td>Nine</td>
<td>92</td>
</tr>
<tr>
<td>أ</td>
<td>28</td>
</tr>
<tr>
<td>خ</td>
<td>24</td>
</tr>
<tr>
<td>هـ</td>
<td>20</td>
</tr>
</tbody>
</table>
5.2 Future Works

i. For future works, must search about additional descriptors to extend the numbers under recognition.

ii. Trying to add additional characteristics for Arabic characters over other languages and try to manipulate it by the computer.

iii. Using more complex neural network architectures with more testing.

iv. This project is done, basically for single numbers, the future works can be done for combined numbers, applied for all Arabic letters, words and sentences.

v. Since creating an algorithm with a one hundred percent correct recognition rate is quite probably impossible in our world of noise and different font styles, it is important to design character recognition algorithms with these failures in mind so that when mistakes are inevitably made, they will at least be understandable and predictable to the person working with the program.
REFERENCES:


[7] Dr. Tahani Abdalla Attia, lecture notes of (Neural Network and Fuzzy Logic), University of Khartoum Faculty of Engineering, 2010.


close all;
clear all;
clc;

sel=[1 7 26];
x= zeros(288,length(sel));
sum= zeros (288,288);
for i=1:length(sel)
    t=imread(sprintf('new letters\%d.png',sel(i)));
    t=rgb2gray(t);
    t=t>0;
    if i==1
        t=ones(18,16);
        t(3:17,6:8)=0;
    end
    if i==2
       t(15,2)=0;
       t(9:10,2:3)=0;
       t(8,5)=0;
    end
    if i==3
       t(5:10,15:16)=1;
       t(15:16,1:14)=0;
       t(8:14,13:14)=0;
       t(9:11,9:12)=1;
       t(12:14,3:4)=0;
       t(13:14,7:8)=0;
       t(9:10,2)=1;
       t(2,7)=1;
       t(8,5)=0;
    end
    t=1-t;
    x(:,i)=2*t(:)-1;
end
r=x(:,3);
tt= reshape(r,18,16);
colormap([1 1 1;0 0 0])
%tt(1:2,:)=-1;
imagesc(tt)

w= x'*x' - length(sel) * eye(288);
save('data.mat','x','w')
Hopfield Algorithm

close all;
clear all;
clc;
colormap([1 1 1; 0 0 0])
load data;
load letters;
% tested on 5aa 8,12,13
%tt=imread('PR\5aa\12.png');
% tested on alf 3,4,5,6,7,8,9
% tt=imread('PR\alf\9.png');
% tested on haa
%tt=imread('PR\haa\25.png');

%tt=rgb2gray(tt);
%tt=tt(70:183,85:150);
%imagesc(tt)
%pause(0.5)
%tt = imresize(tt, [18 16]);
%tt=tt>250;
%tt=1-tt;
%tt=2*tt-1;
%tt= reshape(tt,18,16);
%imagesc(tt)
%r=tt(:);
%pause
F=1;
while (F)
  F=0;
  %j=randperm(288);
  y= sign(w*r);
  if norm(y-r)==0
    F=1;
  end
  j=find(y==r);
  if j>0
    jj=randperm(length(j));
    r(j(jj(1)))=y(j(jj(1)));
  end
  t= reshape(r,18,16);
  colormap([1 1 1; 0 0 0])
  imagesc(t)
  pause(0.1)
end
load handel
sound(y,Fs)
Numbers

Nested IF algorithm

```plaintext
x=input;
threshold;
negative;
discribe;
lowerwidth;
upperwidth;
thewidth;
thehight;
leftdensity;
upperdensity;
linedensity;
readnum;
if(cu==2)
    imshow(seven)
else
    if(cl==2)
        imshow(eight)
    else
        if (((D.Area/48327)<=0.07) && (hight/267)<=0.4)
            imshow(zero)
        else
            if ((width/hight)<=0.4)
                imshow(one)
            else
                if (h==1)
                    if (upper>=0.6)
                        imshow(nine)
                    else
                        imshow(five)
                end
            else
                if ((low>=up)&&(cu==1)&&(cl==1))
                    imshow(four)
                else
                    if (left>=0.6)
                        if (line>=0.2)
                            imshow(two)
                        else
                            imshow(three)
                    end
                else
                    if (left<=0.4)
                        imshow(six)
                end
            end
        end
    end
end
```

A3
Perceptrons and linear neural networks algorithm

```matlab
x=input;
threshold;
negative;
discribe;
lowerwidth;
upperwidth;
thewidth;
thehight;
leftdensity;
upperdensity;
linedensity;
readnum;
A=cu+cl;
if(width<=181/2)
w=0;
else
 w=1;
end
if((h==1)&&(ca==2)&&(w==1))
  net = newp([-2 2],1);
p = [0 1];
t = train(net,p,t);
a = sim(net,uh);
  if (a==1)
   imshow(nine)
  else
   imshow(five)
end
end
if((h==0)&&(ca==3)&&(w==1))
  net = newp([-2 2],1);
p = [2 1];
t = [0 1];
 net = train(net,p,t);
a = sim(net,cu);
  if (a==1)
   imshow(eight)
  else
   imshow(seven)
end
else
   imshow(seven)
end
end
if((h==0)&&(ca==3)&&(w==0))
  net = newp([-2 2],1);
p = [0.4 0.1];
t = [0 1];
net = train(net,p,t);
a = sim(net,r);
  if (a==1)
   imshow(one)
  else
   imshow(zero)
end
end
if((h==0)&&(ca==2)&&(w==1))
  net = newlin([-2 2; -2 2],1);
P = [0.6 0.7 0.2 ;0 1 0];
t = [0 1 2];
net.trainParam.goal = 0.1;
```
[net, tr] = train(net, P, t);
A = sim(net, Q);
if (A <= 0.5)
    if (line >= 0.2)
        imshow(two)
    else
        imshow(three)
    end
else
    if (A <= 1 && A >= 0.5)
        imshow(four)
    else
        if (A >= 1)
            imshow(six)
        end
    end
end
Threshold

```
[a b c]=size(x);
for i= 1:1:a
    for j= 1:1:b
        y(i,j)=x(i,j);
    end;
end;
d=255*graythresh(y);
[a b]=size(y);
for i=1:1:a
    for j=1:1:b
        if y(i,j)>d
            f(i,j)=255;
        else
            f(i,j)=0;
        end;
    end;
end;
```

Describe

```
[L c]=bwlabel(g);
D=regionprops(L,'PixelList','EulerNumber','Extrema','Area','BoundingBox','Centroid');
h= c - D.EulerNumber;
for p=1:133
    for o=1:181
        m(p,o)=g(p,o);
    end
end

for p=134:267
    for o=1:181
        n(p-133,o)=g(p,o);
    end
end
```
Appendix A

**THE CODE**

**Left density**

```
l=0;
for k = 1:267
    for j= 1:91
        if(g(k,j)==255)
            l =l+1;
        end
    end
end
left=l/D.Area;
if((cu==1)&&(cl==1))
    if(low>=up)
        q=1;
    else
        q=0;
    end
Q=[left; q];
end
```

**Line density**

```
l=0;
for k = round(D.Extrema(2,2)):round(D.Extrema(2,2)+20)
    for j= 1:181
        if(g(k,j)==255)
            l =l+1;
        end
    end
end
line=l/D.Area;
```

**Lower width**

```
[H cl]=bwlabel(n);
N=regionprops(H,'PixelList','EulerNumber','Extrema','Area','BoundingBox','Centroid');
if(cl==1)
    low = N.Extrema(4,1) - N.Extrema(7,1);
end
```

**Negative**

```
for i=1:1:a
    for j=1:1:b
        if(f(i,j)==0)
            g(i,j)=255;
        else
            g(i,j)=0;
        end
end
end
```
Appendix A

Read numbers

one= imread('C:\Users\Packard bell\Desktop\one.png');
two= imread('C:\Users\Packard bell\Desktop\two.png');
three= imread('C:\Users\Packard bell\Desktop\three.png');
four= imread('C:\Users\Packard bell\Desktop\four.png');
five= imread('C:\Users\Packard bell\Desktop\five.png');
six= imread('C:\Users\Packard bell\Desktop\six.png');
seven= imread('C:\Users\Packard bell\Desktop\seven.png');
eight= imread('C:\Users\Packard bell\Desktop\eight.png');
nine= imread('C:\Users\Packard bell\Desktop\nine.png');
zero= imread('C:\Users\Packard bell\Desktop\zero.png');

The hight

hight=D.BoundingBox(4);
r= width/hight;

The width

width= D.BoundingBox(3)

Upper density

i=0;
for k = 1:130
   for j= 1:181
      if(g(k,j)==255)
         i =i+1;
      end
   end
end
upper=i/D.Area;

Up density

[F cu]=bwlabel(m);
M=regionprops(F,'PixelList','EulerNumber','Extrema','Area','BoundingBox','Centroid');
if(cu==1)
   up= M.Extrema(3,1)-M.Extrema(8,1);
   end
if(cu==1)
uh= cu - M.EulerNumber;
   end
ca= cl+cu;
University of Khartoum

Faculty of engineering

Electrical department

Final year

A research on **characters recognition**

Please rewrite the following character in ARABIC language, CLEARLY, CENTERED and in good WIDTH.

a) Numbers from (0 to 9) "in Arabic"

b) Arabic Letters (٢، خ، ١)

Thanks a lot for helping us......

By: Lowai Kamal El-Deen & Abd El-Rahman Al-Tegany
Some Samples:

University of Khartoum
Faculty of engineering
Electrical department
Final year

A research on characters recognition

Please rewrite the following character in ARABIC language.
CLEARLY, CENTERED and in good WIDTH.

a) Numbers from (0 to 9) "in Arabic"

b) Arabic Letters 

Thanks a lot for helping us....

by: Lowal Karral Al-deen & Abd Al-Rahman Al-Tayyib
University of Khartoum
Faculty of engineering
Electrical department
Final year
A research on characters recognition

Please rewrite the following character in ARABIC language, CLEARLY, CENTERED and in good WIDTH.

a) Numbers from (0 to 9) “in Arabic”

\[
\begin{array}{ccc}
1 & 5 & 3 \\
3 & 0 & 7 \\
\Lambda & \Lambda & 9 \\
0 & & \\
\end{array}
\]

b) Arabic Letters (٦, ٩)

Thanks a lot for helping us.....

by: Lowal Kamal Al-deen & Abd Al-Rhman Al-Tegany