جمهورية العراق
وزارة التعليم العالي و البحث العلمي
جامعة النهرين
كلية العلوم

تميز ألاحرف باستخدام تحويلات ألصوره ألهجينه

رسالة مقدمة الى كلية العلوم، جامعة النهرين
كجزء من متطلبات نيل شهادة الماجستير في علوم الحاسوب

من قبل
هبه عبد الرضا عبد الامير علي أغا

بكالوريوس
٢٠٠٣

المشرف
د. عبد الكريم عبد الرحمن كاظم

شباط ٢٠٠٨ صفر ١٤٢٩

Republic of Iraq

Ministry of High Education & Scientific Research

Al-Nahrain University

Collage of Science

Character Recognition
Using Hybrid Image Transform

A THESIS

SUBMITED TO THE

COLLEGE OF SCIENCE / NAHRAIN UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

IN COMPUTER SCIENCE

By

HEBA ABDULREDA A-A. ALI AGHA

(B.Sc 2003)

Supervisor

DR. ABDULKARIM A-R. KADHIM

February 2008 Sofar 1429

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Computer Science / Collage of Science / Al-Nahrain University, by **HEBA ABDULREDA A-A. ALI AGHA** as partial fulfillment of the requirement for the degree of Master of Science in Computer Science.

Signature:
Name: **Dr. AbdulKarim A-R. Kadhim**
Title: Assistant Professor
Date: / / 2008

In view of the available recommendation, I 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: 17 / 2 / 2008

Examining Committee Certification

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.

Examining Committee Certification

Signature:
Name: **Abdul Monem S. Rahma**
Title: **Assist. Prof. (Chairman)**
Date: / / **2008**

Signature:
Name: **Dr. Jane Jaleel Stepha**
Title: **Assist. Prof. (Member)**
Date: / / **2008**

Signature:
Name: **Dr. Ban N. Al-Kallak**
Title: **Lecturer (Member)**
Date: / / **2008**

Supervisors Certification

Signature:
Name: **Dr. AbdulKarim A-R. Kadhim**
Title: **Assist. Prof.**
Date: / / **2008**

The Dean of the College Certification

Approved by the Council of the College of Science

Signature:
Name: **Dr. LAITH ABDUL AZIZ AL - ANI**
Title: The Dean of College of Science**, Al-Nahrain University.**
Date: / / **2008**

الاهداء

إلى من زرع في نفسي الطموح و الأمل و العطاء...
إلى من ذلل لي المصاعب وهون عليّ المتاعب...
إلى مثلي الأعلى في الحياة...
أبي

إلى من سهرت من أجلي الليالي ورخصت لي الغالي...
إلى الشمعه التي تحترق لتضيء لنا الطريق...
إلى من خصتني كل يوم بالدعاء...
أمي

إلى أرق و أحن و أطيب انسان...
إلى سندي في الحياة...
إلى نصفي الثاني و شريك حياتي ...
زوجي

إلى من شد الله بهم أزري ...
و ساعدني بهم على تخطي مِحني...
أخواتي

أهدي ما وفقني اليه ربي

هبه عبد الرضا علي أغا

Acknowledgment

First of all I would like to express my sincere gratitude and appreciation to my supervisors DR. ABDULKARIM A-R. KADHIM.

Grateful thanks to DR. HATHIM A. OMAR for his valuable guidance, supervision.

Special thanks to the College of Science, dean of the college for the continuous supports and encouragement during the period of my studies.

Grateful thanks for the Head of Department of Computer Science Dr. TAHA S. BASHAGA, staff and employees.

Finally, my very special thanks to my family, my husband, and my friends especially, M.Sc. group for their continuous supports and encouragement during the period of my studies.

Heba

Chapter Five

Conclusion and Future Work

5.1 Conclusion

The following remarks represent the main work conclusion;

1. The preprocessing operation has a tremendous effect on the recognition rates obtain by the system.
2. The recognition rates obtained in the work show that the (DCT) had the highest recognition rate presented, at a relative high number of coefficient.
3. The computational time of DCT is quite large compared with WT, which consider as a weakness for the DCT. That means the DCT can be preferred for offline applications, but WT preferred for online applications.
4. The results show that Daubechies WT (Db4) gives better recognition rates in comparison with Haar WT. This is true with different levels tested in the work (two, three and four levels).

5.2 Future Work

Many suggestions are possible for future work;

1. Modify the proposed recognition system for complete words (without spaces).

2. Modify the proposed recognition system to be used with different types of printed fonts and character size and with handwritten characters.
3. Modifying the recognition system to work with Arabic characters.
4. Using Wavelet transformation as feature extraction techniques followed by more sophisticated methods for decisions such as neural based methods.
5. To overcome the problem of low recognition rates for some of the characters in both feature extraction techniques (DCT, Haar WT and Db4 WT), applying both techniques might improve the recognition rates.

Chapter Four

Test Results and Discussion

4.1 Introduction

This chapter is concerned the discussion of the results produced by the proposed scheme to perform the operation of recognizing printed characters for English letters (capital & small) and decimal digits. Calculations of the mentioned results in this chapter are based on 62 samples for training and 10 bmp files for testing which includes **744** samples in addition to 62 bmp files for testing each letter or digit separately which includes **620** samples. The training and test samples are scanned at 200 dpi resolution of 256 gray-scales.

The programming language used for this recognition system is Visual C^{++} 6.0.

4.2 System Training Process

The training file used contains characters (capital & small letters) and the digits (0 to 9), so the total number of samples in data base (DB) is given by:

Total DB Samples = capital letters + small letters + 10 digits

= 26 + 26 + 10

= 62 samples

In the building the database phase, feature extraction method (DCT or WT) is performed on the segmented preprocessed image then storing those features in an external file as a reference in the DB.

4.3 System Testing Process

Ten bmp files where built to be used for testing (the files contain a total of 744 different samples of the characters to cover each small and capital letters and each digit twelve times), as shown in Appendix B:

$$744 = 26(\text{characters}) \times 2(\text{capital \& small}) \times 12(\text{repeating times}) \\ + 10(\text{digits 0-9}) \times 12(\text{repeating times})$$

The file where used to calculate the percentage of recognition rates to be able to compare the results which are given by the different feature extraction techniques (DCT, Haar WT and Db4 WT) adapted in this research. DCT or wavelet features are extracted from each file separately then Euclidean distance is computed as the difference between the test pattern feature and the training patterns features in database. Finally, the recognition rate for the proposed system is computed as the percentage of successful recognitions as given by Equation (3.3).

For testing separate symbols (characters and digits) 26 files where used, each file contains one of the symbols repeated 10 times. With this testing the recognition percent of each symbol where calculated.

4.4 Experimental Results

In this section, a number of tests were performed to study the system behavior with the feature extraction methods used in the recognition system.

4.4.1 Recognition using DCT

Table (4.1) illustrates the recognition rates for DCT technique on the 10 testing bmp files (Appendix B) with different number of coefficients.

Table (4.1) The Recognition Rates using DCT

Sample files	**DCT Coeff. I=1**	**DCT Coeff. I=5**	**DCT Coeff. I=10**	**DCT Coeff. I=15**	**DCT Coeff. I=20**	**DCT Coeff. I=25**	**DCT Coeff. I=30**
Test1.bmp	7.258	45.967	74.193	86.29	91.129	93.548	93.548
Test2.bmp	21.774	77.419	87.903	95.161	95.967	97.580	98.387
Test3.bmp	14.516	70.967	82.258	85.484	91.935	93.548	93.548
Test4.bmp	20.967	72.580	83.871	88.709	93.548	96.774	96.774
Test5.bmp	19.354	67.741	83.871	91.935	93.548	95.161	96.775
Test6.bmp	29.032	75.806	87.096	91.935	93.548	95.161	95.161
Test7.bmp	19.354	72.580	88.709	95.161	98.387	98.387	98.387
Test8.bmp	19.354	64.516	83.871	87.096	91.935	96.774	96.774
Test9.bmp	20.967	70.967	87.096	95.161	95.161	95.161	93.548
Test10.bmp	16.129	69.354	83.871	91.935	91.935	98.387	98.387
ARR	18.8705	68.7897	84.2739	90.8867	93.7093	96.0481	96.1289

Where:

I: represents the number of DCT Coefficients

$$\text{Average Rercognition Rate}\left(\text{ARR}\right) = \sum_{i=1}^{N} \text{Recognition Rate Test Sample} / \text{N}$$

N: Number of testing images.

From the results shown in the table, it is clearly seen that a number of coefficients of 25 gives a reasonable recognition rates. It has been seen that increasing the used number of coefficients beyond 25 does not improve the recognition rates significantly. Increasing the number of coefficients accordingly increase the computational time. Thus ARR=96.0481%, so this level is taken to calculate the recognition rate for each sample, characters (capital and small) and digits (0-9), which was illustrated in table (4.2), the testing files contains each symbol separately on an image repeated 10 times but with only one training sample.

From table (4.2) one can observe that this feature extraction method is good and suitable for most of the symbols used in this research.

Table (4.2) Recognition rates using DCT with 25 coefficients

SAMPLE	RECOGNITION RATE	SAMPLE	RECOGNITION RATE
A	100	5	70
B	80	6	100
C	100	7	70
D	100	8	100
E	100	9	70
F	100	a	80
G	100	b	100
H	100	c	80
I	100	d	100
J	100	e	50
K	100	f	100
L	100	g	70
M	100	h	100
N	100	i	90
O	80	j	80
P	100	k	100
Q	100	l	100
R	80	m	100
S	100	n	100
T	100	o	70
U	100	p	100
V	100	q	100
W	100	r	100
X	100	s	100
Y	100	t	90
Z	100	u	100
0	100	v	100
1	100	w	100
2	70	x	100
3	100	y	70
4	100	z	90

4.4.2 Recognition using Haar WT

Table (4.3) demonstrates the recognition rates for Haar Wavelet Transform Technique according to different wavelet decomposition levels.

Table (4.3): The Recognition Rates using Haar WT

Sample files	Haar Coeff. 1-Level	Haar Coeff. 2-Levels	Haar Coeff. 3-Levels	Haar Coeff. 4-Levels
Test1.bmp	15.322	25.806	45.161	66.935
Test2.bmp	32.258	38.709	59.677	86.290
Test3.bmp	22.580	32.258	62.903	82.258
Test4.bmp	22.580	29.032	67.741	85.483
Test5.bmp	24.193	24.193	58.064	88.709
Test6.bmp	25.806	35.483	59.677	85.483
Test7.bmp	25.806	29.032	53.225	83.871
Test8.bmp	24.193	30.645	51.612	83.871
Test9.bmp	30.645	35.483	67.741	96.774
Test10.bmp	32.258	40.322	67.741	88.709
ARR	25.5641	32.0963	59.3542	84.8383

Where:

$$\text{Average Rercognition Rate}\left(\text{ARR}\right) = \sum_{i=1}^{N} \text{Recognition Rate Test Sample} / \text{N}$$

N: number of test sample.

From the above table, the obtained results obviously indicate that four wavelet decomposition levels produce the best recognition rates for Haar WT, with 13 coefficients. With ARR=84.83%, so this level is taken to calculate the recognition rate for each sample, characters (capital and small) and digits (0-9) which was illustrated in table (4.4), the testing files contains each symbol separately on an image repeated 10 times but with only one training sample.

Table (4.4) Recognition rate using 4-level Haar WT decomposition:

SAMPLE	RECOGNITION RATE	SAMPLE	RECOGNITION RATE
A	100	5	20
B	100	6	100
C	90	7	60
D	100	8	90
E	10	9	50
F	80	a	100
G	70	b	30
H	100	c	90
I	80	d	30
J	20	e	100
K	100	f	80
L	90	g	20
M	90	h	10
N	90	i	40
O	90	j	10
P	70	k	10
Q	100	l	90
R	100	m	100
S	70	n	100
T	100	o	50
U	70	p	70
V	100	q	70
W	100	r	60
X	90	s	100
Y	60	t	100
Z	80	u	100
0	100	v	100
1	60	w	90
2	20	x	100
3	100	y	20
4	100	z	90

4.4.3 Recognition Rates using Daubechies-4 WT (Db4)

Table (4.5) illustrates the recognition rates for Db4 Wavelet Transform Technique according to different wavelet decomposition levels.

Table (4.5): The Recognition Rates using Daubechies WT

Sample files	Db4 Coeff. 1-Level	Db4 Coeff. 2-Levels	Db4 Coeff. 3-Levels	Db4 Coeff. 4-Levels
Test1.bmp	38.709	41.129	60.483	76.612
Test2.bmp	51.612	62.903	75.806	91.935
Test3.bmp	48.387	66.129	75.806	88.709
Test4.bmp	41.935	56.451	74.193	93.548
Test5.bmp	54.838	66.129	77.419	91.935
Test6.bmp	41.935	54.838	77.419	91.935
Test7.bmp	38.709	56.451	72.580	88.709
Test8.bmp	41.935	64.516	77.419	93.548
Test9.bmp	61.290	62.903	79.032	96.774
Test10.bmp	48.387	61.290	80.645	93.548
ARR	46.7737	59.2739	75.0802	90.7253

Where:

$$\text{Average Rercognition Rate}\left(\text{ARR}\right) = \sum_{i=1}^{N} \text{Recognition Rate Test Sample} / \text{N}$$

N: number of test samples.

From the results obtained from the above table, 4-level wavelet produces the best recognition rates for Db4 WT. With ARR=90.7253%, so this level is taken to calculate the recognition rate for each sample, characters (capital and small) and digits (0-9) which was illustrated in table (4.6), the testing files contains each symbol separately on an image repeated 10 times but with only one training sample.

Table (4-6) Recognition rate using 4-level Daubechies WT

SAMPLE TEST	RECOGNITION RATE	SAMPLE TEST	RECOGNITION RATE
A	60	5	20
B	100	6	100
C	100	7	40
D	100	8	100
E	10	9	50
F	100	a	100
G	90	b	70
H	100	c	80
I	80	d	100
J	90	e	90
K	90	f	70
L	20	g	20
M	90	h	20
N	100	i	90
O	80	j	10
P	90	k	100
Q	90	l	100
R	20	m	100
S	90	n	50
T	100	o	70
U	90	p	100
V	100	q	100
W	100	r	100
X	100	s	70
Y	100	t	80
Z	100	u	100
0	100	v	90
1	100	w	100
2	20	x	90
3	80	y	20
4	100	z	40

4.5 Discussion

In this research, images in Appendix B were firstly designed by the paint and stored as digital image, these images were used first to test the system, which gives result between 99.5% to 100% accuracy. But these digital images were printed and scanned and then used to test the system, which gives the obtained result mentioned in the given tables in this chapter.

From the previous tables, one can conclude the followings:

1. The quality of printer, the resolution of the scanner, and the quality of paper have high effects on the recognition rates.
2. From Table (4.1), a number of coefficients of 25 seems reasonable to get high recognition rates for DCT method.
3. 4-level of wavelet decomposition produces the best recognition rates for both WT methods (Haar and Db4), as shown in Table (4.2) and Table (4.3).
4. Table (4.3) and table (4.5) shows that Daubechies WT gives better recognition rates than Haar WT.
5. For all the methods used in the work, the best recognition rates are obtained using DCT technique. With 25 coefficients which require relatively more computational time.
6. For both feature extraction techniques used in this research the recognition rates for some characters are significantly low.

Chapter one

Introduction

1.1 Motivation

Recognition regarded as a basic attribute of human beings and other living creatures. According to the nature of objects to be recognized, the process of recognition can be divided into two major types: the recognition of concrete items and the recognition of abstract items. Pictures, characters, music and the object can be recognized around us. This process may be referred to as *sensory recognition,* which includes visual and aural pattern recognition. On the other hand, one can recognize an old argument or a solution to a problem without resorting to external stimuli. This process involves the recognition of abstract items and can be termed *conceptual recognition* [Ema95].

1.2 Pattern Recognition

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes. Depending on the application, these objects can be images, signal waveforms or any type of measurements that need to be classified [Ser03], figure (1.1) illustrate the pattern recognition family, which divided mainly into sound recognition and image recognition.

The principle function of a pattern recognition system is decision making concerning the class membership of the patterns, and with which classes to be compared.

A pattern can be defined as a quantitative or structural description of an object or some other entity of interest, while a pattern class is a set of patterns that share some common characteristics [Ema95].
In general, pattern recognition systems have general style of work and have common phases in order to perform its requiring job.

The image recognition is part of the pattern recognition and the text recognition is part of the image recognition. In general, character by character recognition is need to made text recognition.

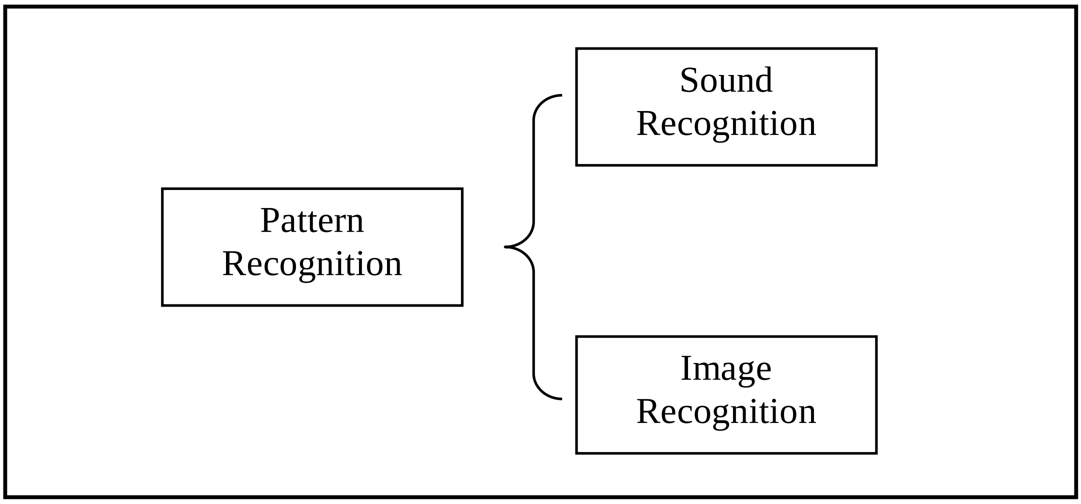


Figure (1.1) Pattern Recognition Family

1.2.1 Text Recognition

Text recognition problem is widely studied in pattern recognition field for the last two decades because of both it theoretical value in pattern recognition and its numerous application like automatic processing of post addresses in mail letters, automatic money amounts determination in bank checks, processing and analyzing of printed text documents [Mac96].

The problem of printed text processing is of great importance because its satisfactory solution will allow digitizing millions of printed text and handwritten materials all over the world, thus making them broadly available. For example the problem for reliable and secure preservation of the cultural heritage is especially important and urgent one. This is why so many researchers all over the world have been involved in during the last decades. To implement text recognition, character recognition must be done first.

1.3 Character Recognition

Character recognition is more widely known as optical character recognition (OCR), since it deals with optically processed characters rather than magnetically processed one. The main objective of character recognition is the conversion of a graphical document into a textual one [Tap90].

Various approaches, system architectures and methodologies have been proposed to solve the problem of character recognition [Par99].

Figure 1.2 shows different families of character recognition. Two different families are included in the general term of character recognition [Imp91]:

- On-line character recognition
- Off-line character recognition

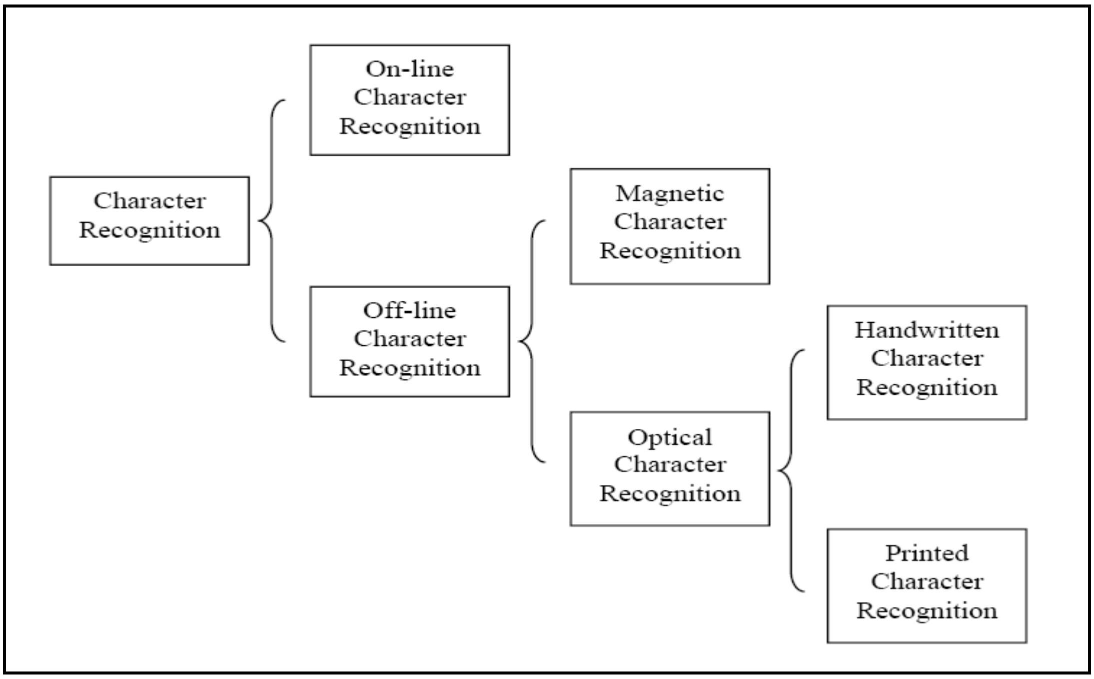

Character Recognition
On-line Character Recognition
Off-line Character Recognition
Magnetic Character Recognition
Optical Character Recognition
Handwritten Character Recognition
Printed Character Recognition

Figure 1.2: The different families of character recognition

On-line character recognition deals with a data stream which comes from a transducer while the user is writing. The typical hardware to collect data is a digitizing tablet which is electromagnetic or pressure sensitive. When the user writes on the tablet, the successive movements of the pen are transformed to a series of electronic signal which is memorized and analyzed by the computer [Tap90].

Off-line character recognition is performed after the writing is finished. The major difference between on-line and off-line character recognition is that on-line character recognition has time-sequence contextual information but off-line data does not. This difference generates a significant divergence in processing architectures and methods.

The off-line character recognition can be further grouped into [Web1]:

- Magnetic character recognition (MCR)

- Optical character recognition (OCR)

In MCR, the characters are printed with magnetic ink. The reading device can recognize the characters according to the unique magnetic field of each character. MCR is mostly used in banks for check authentication.

OCR deals with the recognition of characters acquired by optical means, typically a scanner or a camera. The characters are in the form of pixelized images, and can be either printed or handwritten, of any size, shape, or orientation [Pav92].

The OCR can be subdivided into handwritten character recognition and printed character recognition. Handwritten character recognition is more difficult to implement than printed character recognition due to the diversified human handwriting styles and customs. In printed character recognition, the images to be processed are in the forms of standard fonts like *Times New Roman, Arial, Courier,* etc [Pla02].

Character recognition schemes, which can be divided into four groups as follows:

1. ***Fixed – font recognition:*** which is the recognition of specific font's type written characters. The two fonts in general use are known as OCR-A and OCR-B, illustrated in Fig (1-1) OCR-A was an attempt at creating a stylized set of character which brought together a number of buffering fonts into a unifies standard. OCR-B was designed to produce a set of character with visually acceptable shapes that were as near as possible to conventional character [Awc95].

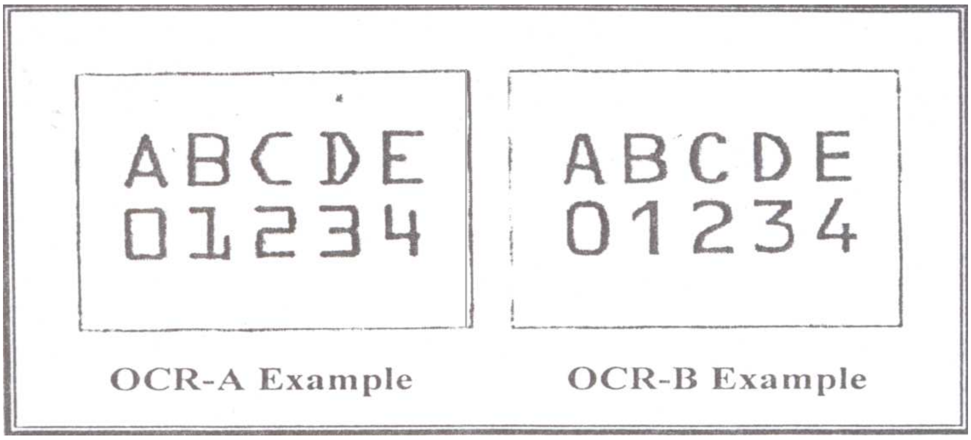


Figure (1.3): OCR-A and OCR-B examples

2. ***On-line recognition:*** which is the recognition of single hand drawn character where not only the character image is provided but also the timing information of each stroke. There have been a large number of research results about On-line character recognition [Bon97].

3. ***Hand – written character recognition:*** which is the recognition of single hand-drawn character of an alphabet, which is unconnected and not written in calligraphy.

4. ***Script recognition:*** It is the recognition of unrestricted handwritten character that may be connected and cursive. Cursive script recognition systems can rely essentially on three types of knowledge: morphological, pragmatic and linguistic. *Morphological* knowledge refers to everything that is known about the shapes of cursive letters. *Pragmatic* knowledge refers to what is known about how to spatially arrange cursive letters into words, phrases and paragraphs. *Linguistic* knowledge

concerns the language that is used to convey the message represented in handwriting [Mar94].

1.4 Problem definition of character recognition

Character recognition is subset of pattern recognition system that is regarded as a system, whose input is the information of the character to be recognized, and the output is a class to which the entered character belong [Lee96].

The problem can be formulated as follows: given a set of N known characters and a new input character. The task is to find out which of the known characters is closest to the new character. This generic problem has many applications, for example, printed character recognition and hand written character recognition. Text recognition problems are usually characterized by a large domain space. For example, recognizing printed characters is very important cause of the huge number of books, sheets and printed materials available all over the world. The task is much more difficult when font-invariant, scale-invariant, shift-invariant, rotation-invariant, or noise-invariant characters should be recognized [Nik98].

1.5 Literature Survey

Pattern recognition fields include a number of applications that have been implemented and studied, i.e., Automatic Target recognition, signature verification problem, Fingerprint Image identification.

- E. W. Brown [Bro92] presents the application of Neural Networks to the problem of identifying machine printed characters in an automated manner. In particular, a back propagation net is trained on an eighty-four character font and tested on two other fonts. Its results are compared against results obtained on the same data by a more traditional approach. This work explores the differences between two different optical character recognition (OCR) algorithms: a feature extraction method using traditional artificial intelligence techniques for classification, and a Neural Network approach with virtually no preprocessing. The basic idea of the experiment was essentially to run the identical data through the two different algorithms and note the differences in each run along the way.

- Chai S. K [Cha95] presents the transformation features using Fourier transformation to generate feature set from Fourier coefficients and points to the powerful characteristics of Fourier transformation of being invariant to image rotation and translation, its main disadvantage is the neglected of all spatial features and depends highly on frequency features.

- Covavisaruch N. [Cov00] presents on-line recognition system of Thai alphabetical characters by using geometrical features that are directly extracted from the image (spatial domain); information extracted for each character is encoded and gathered into several groups such as characters width/height ratio and directional code.

- S. M. Maliki [Mal02] presents models of Neural Network are used to recognize printed characters, applying Artificial Neural Network (ANN) of three types, which are:- Kohenen All Classes in One

Network (ACON), Kohenen One Class in One Network (OCON), and learning Vector Quantization(LVQ). Haar Wavelet transformation is used to extract the parametric features of the printed characters, in addition to the Geometrical features such as: Moment and Complex Moment. The system was implemented on a database of 130 samples, 70 samples from the database were used for training, and the all 130 samples were used for testing the system. The system gives about 91% Recognition rates.

- Chaudhuri B. B. [Cha03] used seven moment invariance as geometrical features in developing an OCR machine, the characteristics of moment's invariant and the result of this research seems to be promising to be used for character recognition, noticing that Chaudhuri B. B. uses also Fourier transformation as feature extraction method for achieving of such promising results.

- K. M. Vamsi [Vam04] presents an attempt is made to recognize printed characters by using features extracted using the proposed sector approach. In this approach, the normalized and thinned character image is divided into sectors with each sector covering a fixed angle. The features totaling 32 include vector distances, angles, occupancy and end-points. For recognition, both Neural Networks and Fuzzy logic techniques are adopted. The proposed approach is implemented and tested on printed isolated character database consisting of English characters, digits and some of the keyboard special characters.

- H. Yan [Yan06] present an algorithm which was developed for printed character recognition based on boundary analysis. New boundary smoothing schemes have been proposed, which can reduce noise significantly. The extracted boundary features are invariant to character size and are convenient for recognition. The whole algorithm is based on the chain code of each boundary and no floating operation is involved. The algorithm is tested with sample pages of a phone directory. Experimental results show that this algorithm is highly reliable and very fast and it gives about 94% recognition rates.

1.6 Aim of the work

The aim of the project is to design and implement a software-based system to recognize printed characters; different types of transformation were used to extract the parametric features of the printed characters. The system is to be used for recognizing printed characters for English letters (capital & small) and decimal digits (0-9), printed with Times New Roman font, size 14.

1.7 Thesis Layout

The following are the outline of the thesis contents:

Chapter Two: provides an overview for the theoretical background for characters recognition system.

Chapter Three: Presents the practical part of the thesis and states the algorithms used in constructing the proposed system.

Chapter Four: Presents the results and discusses certain study cases used to test the system.

Chapter Five: Gives some conclusions and suggestions for future work.

Chapter Three

Proposed Character Recognition System

3.1 Introduction

This chapter introduces character recognition system including the training and testing conditions. In addition, the performance of the system in identifying all isolated English Alphabetic (capital & small letters) and isolated digits (0 to 9) written in Times New Roman font with size 14 is used for reason of its standard use in printing books, thesis and sheets, using the possible preprocessing, segmentation, shifting, feature extraction, transformation and matching method.

3.2 Character Recognition model

The proposed character recognition model is illustrated in figure (3.1), which consists of the usual steps in pattern recognition, as described below:

3.2.1 Input Image

Image that contains the text to be recognized is acquired by using scanner with 256 gray scales and size 512×512 will be saved for further processing as shown in figure (3.2).

3.2.2 Preprocessing

Key function of preprocessing is to improve the input image in way that increase the chance for success of the following process, and using different image processing techniques for normalizing input patterns. For character recognition, the preprocessing here will include histogram starching, noise removing, and binarization.

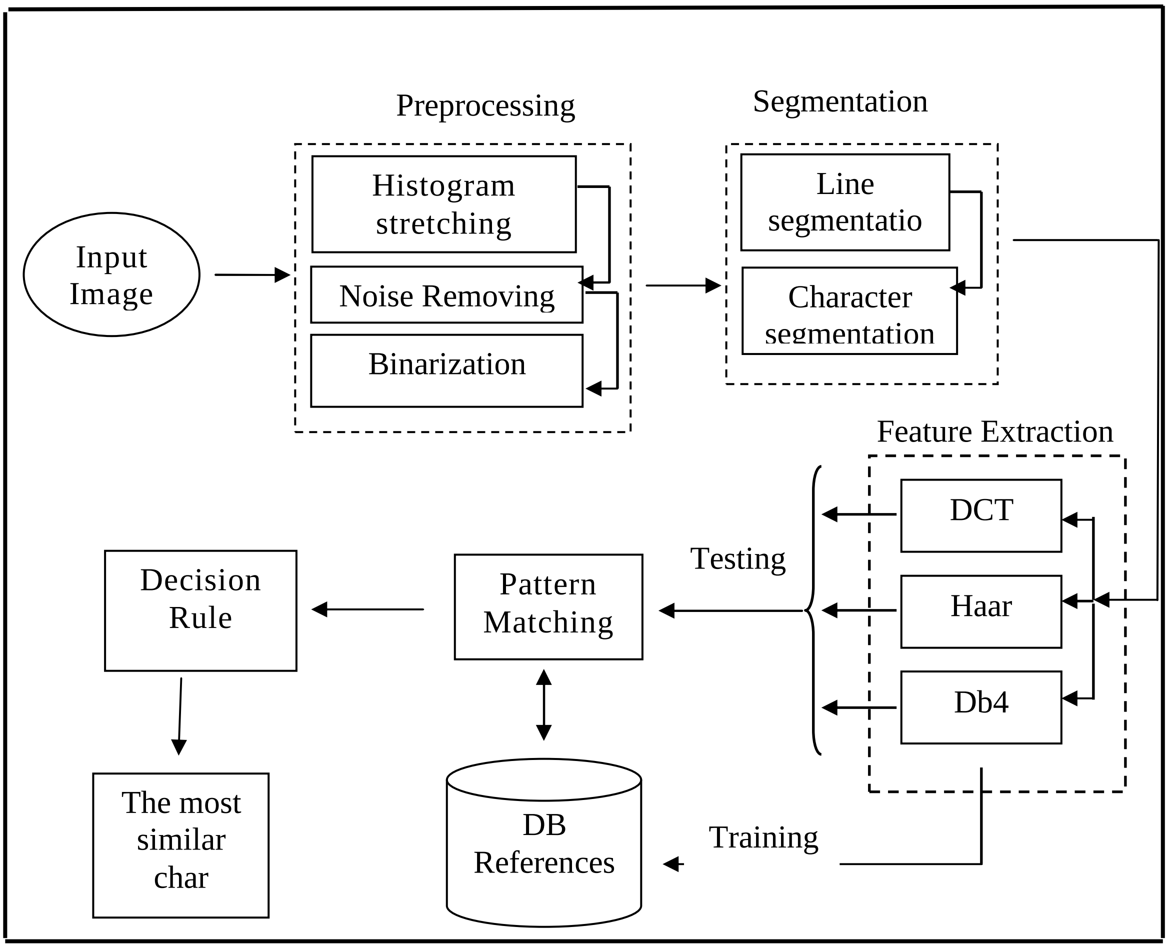
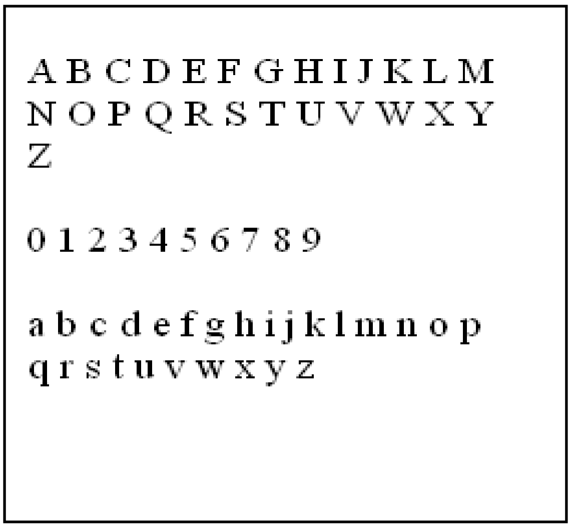


Figure (3.1): The Proposed Character Recognition Model



A B C D E F G H I J K L M
N O P Q R S T U V W X Y
Z

0 1 2 3 4 5 6 7 8 9

a b c d e f g h i j k l m n o p
q r s t u v w x y z

Figure (3.2): The Training Scanned Image

3.2.2.1 Histogram stretching

The histogram stretching is the one of the well-known and the mostly used histogram algorithms to improve a picture. This method selects and stretches the main power interval of the histogram. The calculation of the power interval is selected by a minimal and a maximal point. These points are determinate by a threshold value. After that, the main power range is stretched to the full range (0-255), equation (2.1).

Histogram presents the number of pixels at each brightness pixel in an image. Histogram stretching used to reduce the difference between gray level values in the image.

Algorithm (3.1): Histogram stretching

```
Input:
    I[x][y]: array of pixels (the input image) of size H×W,
    H: height of the image,
    W: width of the image.
Output:
    I[x][y]: array of pixels with stretched value (0-255)
Procedure:
    1. Set Imin is the minimum value of the image=255,
        Set Imax is the maximum value of the image=0.
    2. For all $i \in [0 , H-1], j \in [0 , W-1]$
    3.        Imin = minimum value in the image
    4.        Imax = maximum value in the image
    5. end For
    6. For all $i \in [0 , H-1], j \in [0 , W-1]$
    7.       $I[i][j] = ((I[i][j] - Imin) / (Imax-Imin) \times 255)$
    8. end For
```

Figure (3.3) shows the application of histogram stretching of the original image (before print and scan the image):

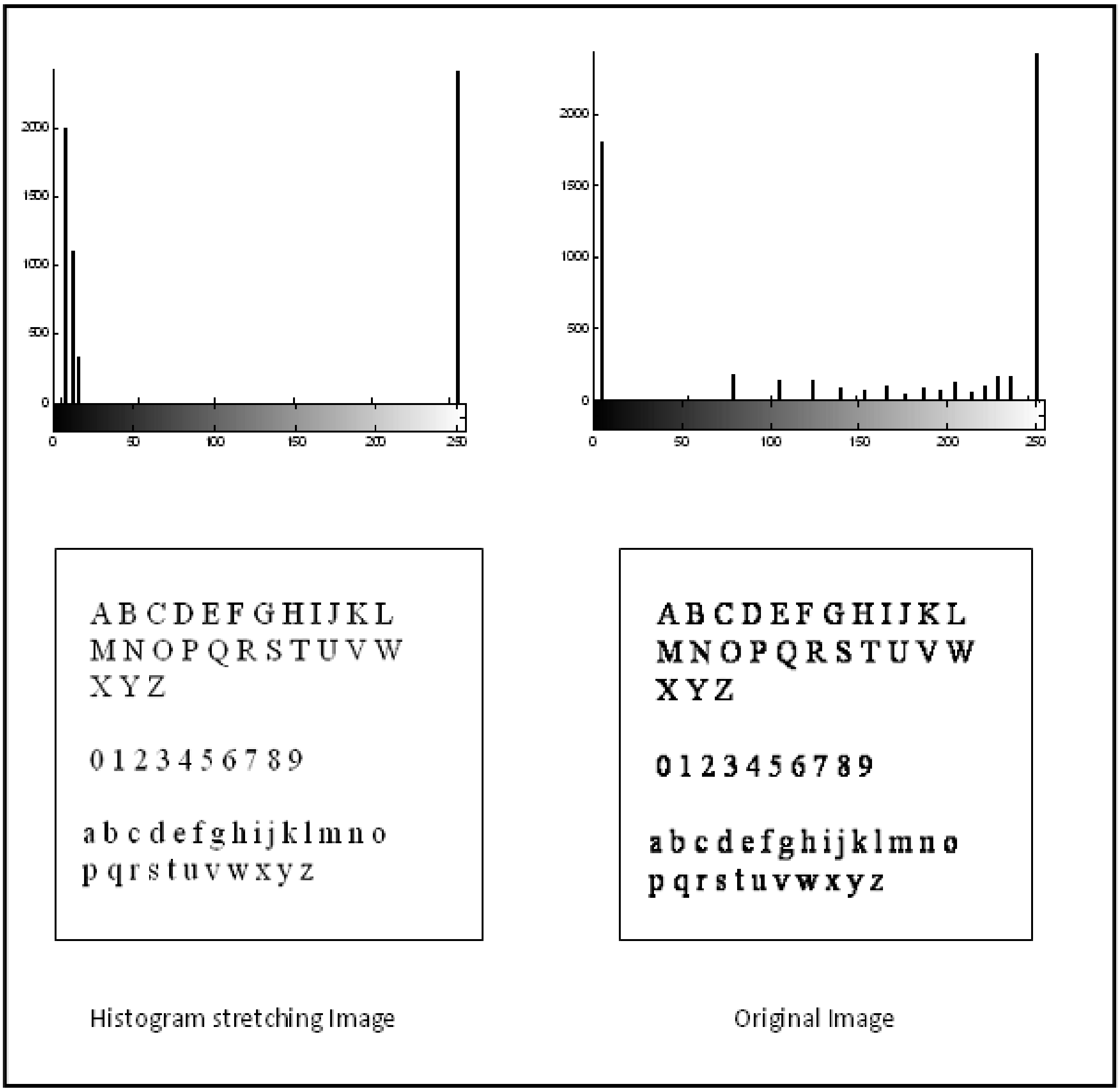


Figure (3.3) The histogram stretching of the original image

Figure (3.4) shows the application of histogram stretching of the scanned image (after print and scan the image):

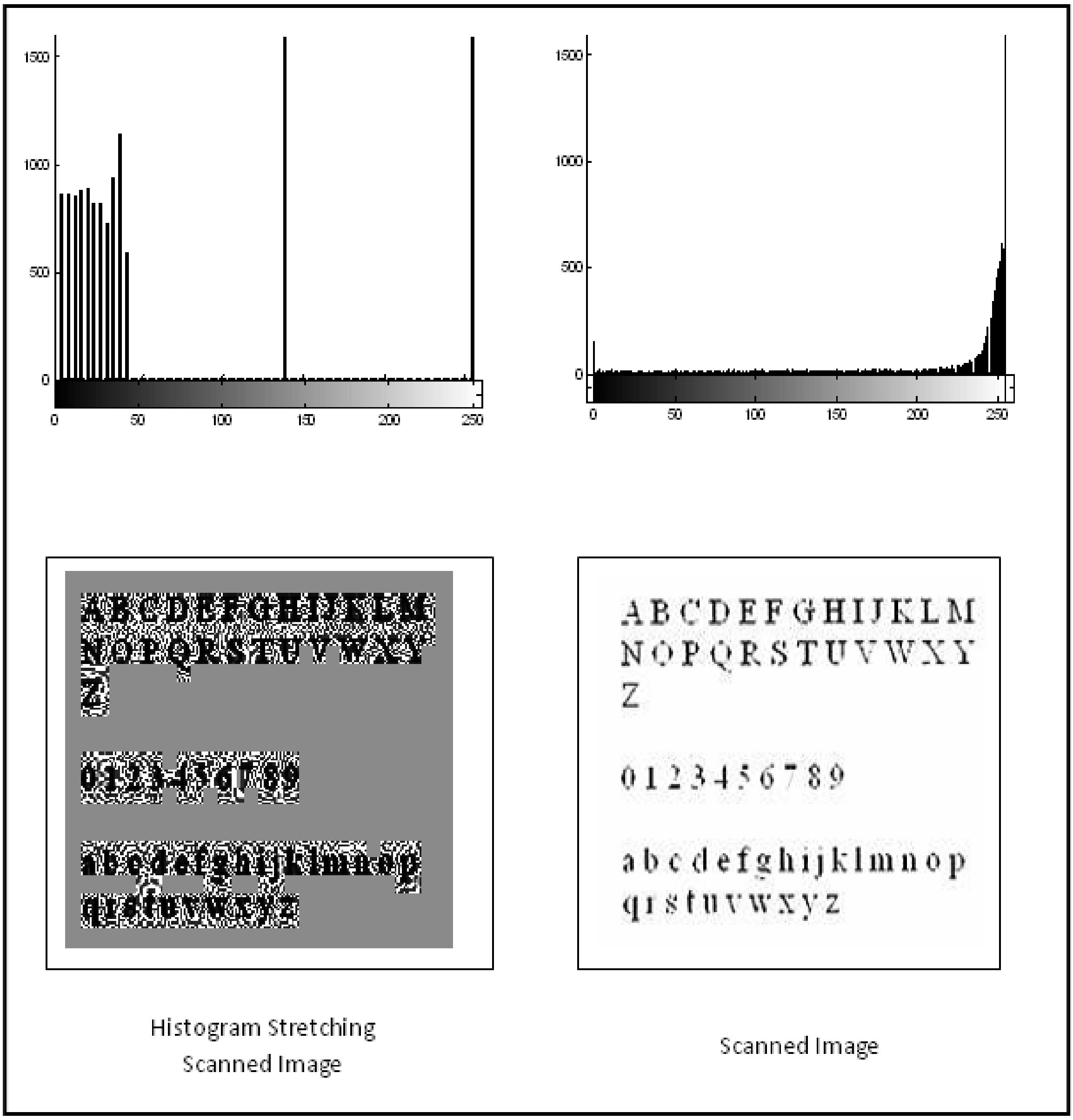


Figure (3.4) the histogram stretching of the scanned image

3.2.2.2 Noise Removing

To remove noise from the input text image Mean filtering noise removing is used, so that the result is more suitable than the original image for specific application. In our work, a 3 × 3 filter size is selected to be convolved with the text image, figure (2.2). The convolution process requires us to overlay the filter on the image, multiply the coincident values, and sum all these results and placing the result of the operation in the location corresponding to center of the filter. When the end of the row is reached, the filter is moved down one row, and the process is repeated row by row until this procedure has been performed on the entire image. Note that the output image must be put in a separate image array, called a buffer, so that the existing values are not overwritten during the convolution process.

Algorithm (3.2): Mean Filter

Input:

I[x][y]: Array of pixels(the input image),

H: High of the image,

W: Width of the image,

Output

I[x][y]: array of noiseless image

Procedure:

```
Set Filtersize=3,
For all i & j < Filtersize
    Mask[i][j]=1/9,
end For
For all i ∈ [0 , H-1], j ∈ [0 , W-1]
    If (i=0) or (i=H-1) or (j=0) or (j=W-1)
        buffer[i][j]=I[i][j],
end For
For all i ∈ [1 , H-1], j ∈ [1 , W-1]
    Set Sumfilter=0,
      For all x ∈ [0 , Filtersize-1], y ∈ [0 , Filtersize-1]
```

```
                Sumfilter=Sumfilter+Mask[x][y]×I[x+i-1][y+j-1],
            end For
            buffer[i][j]=Sumfilter
    end For
    For all i ∈ [0 , H-1], j ∈ [0 , W-1]
        I[i][j]=buffer[i][j],
    end For
```

Figure (3.5) shows the effect of the application of the mean filter on a noisy scanned image (with salt and pepper noise):

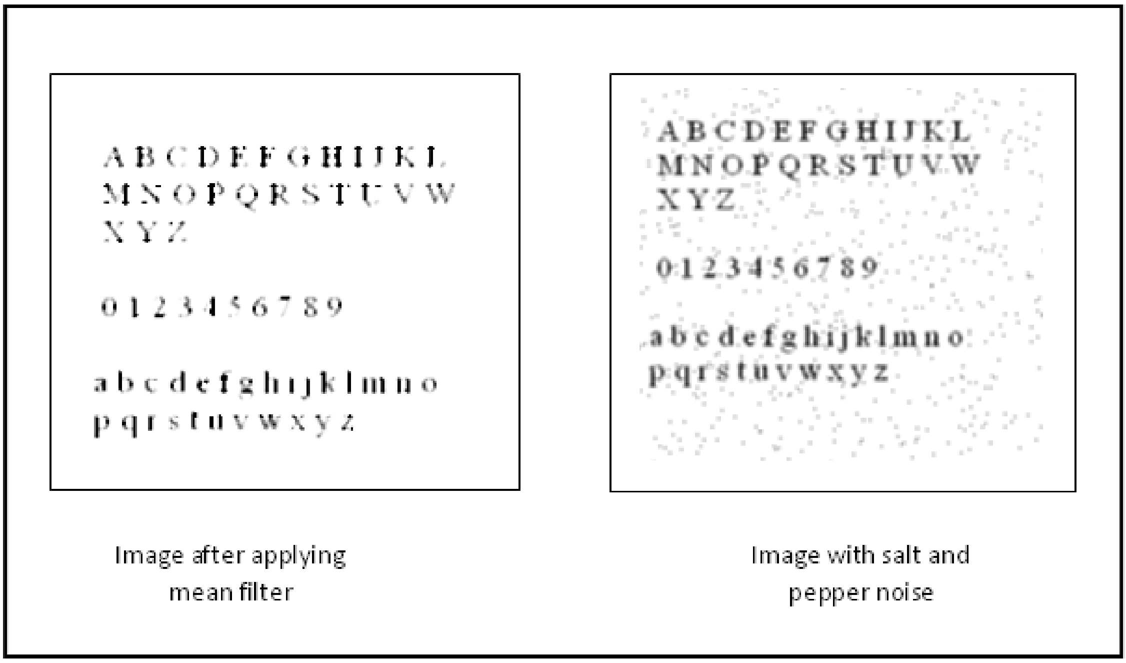


Image after applying mean filter
Image with salt and pepper noise

Figure (3.5) the effect of mean filter on the image

3.2.2.3 Binarization

It is process of turning a gray scale image into a binary image (only two levels 0 and 1) in order to facilitate the feature extraction process. As mentioned in the previous chapter, there are a number of algorithms to find the threshold value, the simplest one use the mean value or the median of the pixel values of the image. All of these algorithms are based on global thresholds or local thresholds.

Global threshold means a value for all the image pixels, and the local threshold means that the image is partitioned into smaller blocks and threshold value is then calculated for each of those blocks a value for a block with fixed size i.e. 32×32 or 16×16, any block size chosen led to the same result, the binarized image. Local thresholds demand a lot more calculations but mostly compensate it with a better result.

Algorithm (3.3): Global threshold binarization

Input:

I[x][y]: Gray scale array of pixels(the input image),

H: Height of the image,

W: Width of the image,

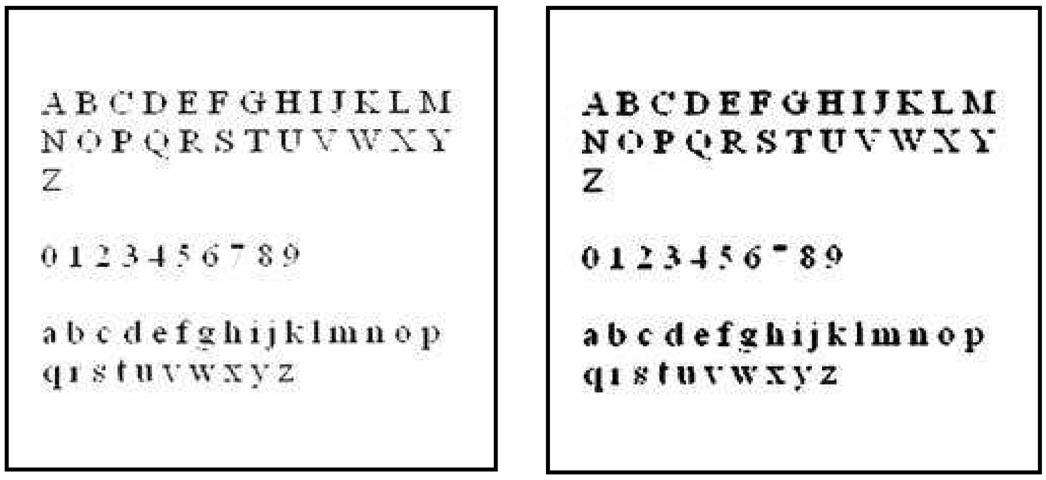
Output

I[x][y]: Binarized image

Procedure

```
1. Set threshold=0,
2. For all i ∈ [0 , H-1], j ∈ [0 , W-1]
3.        threshold=threshold+I[i][j]/(H×W)
4. end For
5. For all i ∈ [0 , H-1], j ∈ [0 , W-1]
6.        If (I[i][j]>threshold) Then (I[i][j]=255)
7.              Else I[i][j]=0
8.  end For
```

Figure (3.4) shows the effect of the application of the binarization technique on the scanned image (after print and scan the image) using local mean threshold value.



A B C D E F G H I J K L M
N O P Q R S T U V W X Y
Z

0 1 2 3 4 5 6 7 8 9

a b c d e f g h i j k l m n o p
q r s t u v w x y z

Scanned image

A B C D E F G H I J K L M
N O P Q R S T U V W X Y
Z

0 1 2 3 4 5 6 7 8 9

a b c d e f g h i j k l m n o p
q r s t u v w x y z

Binarized Scanned image

Figure (3.6) the effect of binarization technique

<u>Algorithm (3.4): Local threshold binarization</u>

Input:

I[x][y]: Array of pixels(the input image),

BlocksizeH: High of the block size,

BlocksizeW: Width of the block size

Output

I[x][y]: Binarized array of pixels.

Procedure

```
1. For all i ∈ [0 , H-1, i=i+BlocksizeH], j ∈ [0 , W-1,j=j+BlocksizeW]
2.       Set threshold=0,
3.       For all x ∈ [0 , BlocksizeH-1], y ∈ [0 , BlocksizeW-1]
```

```
            threshold=threshold+I[x][y]/( BlocksizeH×BlocksizeW)
        end For
        For all x ∈ [0 , BlocksizeH-1], y ∈ [0 , BlocksizeW-1]
            If (I[x+i][y+j]>threshold)   Then  (I[x+i][y+j]=255)
                Else I[x+i][y+j]=0
        end For
    end For
```

3.2.3 Segmentation

Segmentation is defined to be the process of partitioning the image into distinct regions that are meant to correlate strongly with features of interests in the image. Segmentation process is performed by determining the upper and lower bounds for each line of text in the image (i.e., row scan). After that column scan process is performed in order to find the left and right bounds of each character.

In this process, the image is separated into number of sub-images, each of these sub-images contain part of the image that represent an English character (capital or small) instance or a digit (0-9), this image of size 32 × 32 pixel which is suitable symbol size for all the symbols used in this research. Text Image segmentation contains two parts, line segmentation to separate the whole image into lines by finding the upper and lower bound of each line and then character segmentation by finding the left and right bound of each character as shown in figure (3.3) and figure (3.4).

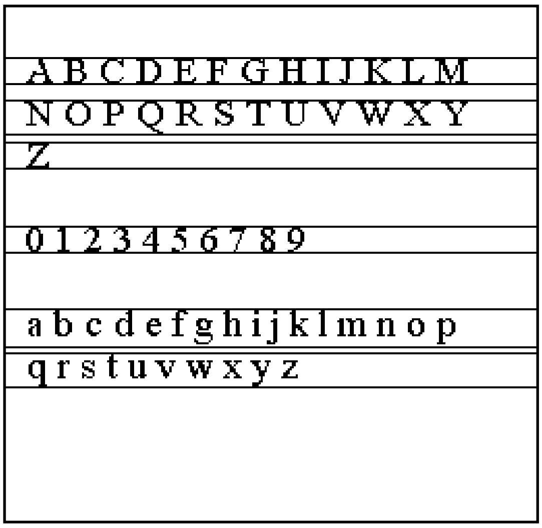


Figure 3.7(3.7): Line Segmentation Image

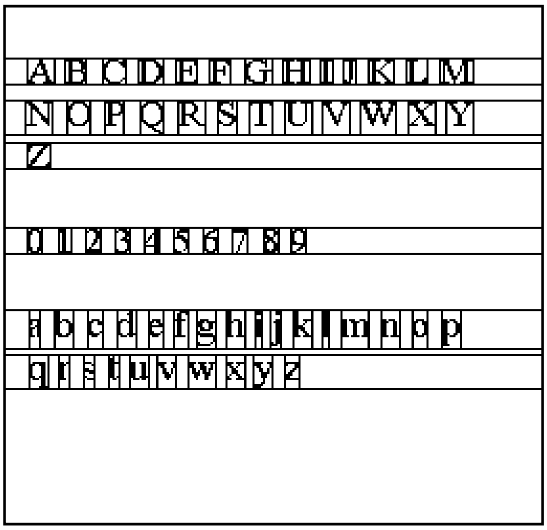


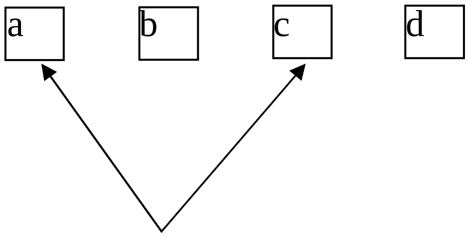
Figure (3.8) Character segmentation Image

In most cases of capital letters and digits it needs only one time row scan because the capital letters have the same height as shown:

And digits also have the same height

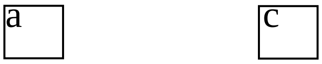
But small letters need another row scan after the column scan since they have different character height.

For example: 'a' 'b' 'c' 'd', where 'a' and 'c' have the same height but of different height than 'b' and 'd'.



One can notice that there is a space on the top of characters 'a' and 'c'

Where in another case it may come with no space like



Then the features will be different which cause a mismatch. So another row scan for each segmented character must be done to deal with this problem.

In the algorithm of line segmentation we defined an array of record named **line**, to be the output array, each cell of it contain two integer values one to get the upper limit and the other to get the lower limit of the line.

Algorithm (3.5): Line Segmentation

Input:

I[x][y]: Array of pixels(the input image),

H: Height of the image,

W: Width of the image,

Output:

Line: array of line segmented image

Procedure:

```
1.  For all i ∈ [0 , H-1]
2.      Line[i].U = -1
3.      Line[i].D = -1
4.  end For
5.  Set Noofline=0
6.  For all i ∈ [0 , H-1], j ∈ [0 , W-1]
7.      If (I[i][j] = 0) then
8.          If (Line[Noofline].U=-1) Then Line[Noofline].U=i
9.              Else If (Line[Noofline].U≠-1) Then Line[Noofline].D=i
                    Noofline=Noofline+1
10. end For
11. Set R=0
12. For all i ∈ [0 , Noofline-2], j ∈ [0 , Noofline-1]
13.     If (Line[i].D = Line[j].U-1)   Then    Line[i].D = Line[j].D
14.         If (j=Noofline-1)    Then      Line[R].U=Line[i].U,
                Line[R].U=Line[i].U,  R=R+1, i=j-1,
15.         Else  Line[R].U = Line[i].U, Line[R].D = Line[i].D, R=R+1,
            i=j-1
16. end For
17. Set Noofline =R
```

Some characters need special attention such as characters i and j because they are contain the dot on their top, if it is not processed the dots will be considered as new line, the following algorithm will deal with this problem.

Algorithm (3.6): Process i, j dot problem

Input:

Line : array of line segmented image.

Nooflines: number of lines in line segmented image.

Output:

ProcessedLine: array of line segmented image(i, j dot processed)

Procedure:

```
1.  Set diff=0, PL=0
2.  For all i ∈ [0 , Noofline-1]
3.      diff=Line[i].D-Line[i].U,
4.      If ( diff <= 2)  then  ProcessedLine[PL].U=Line[i].U,
            ProcessedLine[PL].D=Line[i+1].D, PL++, i++
5.      Else  ProcessedLine[PL].U=Line[i].U,
            ProcessedLine[PL].D=Line[i].D, PL++
6.  end For
7.  Nooflines = PL.
```

In the algorithm of character segmentation we defined an array of record named **CharBorder** to be the output array, each cell of it contain four integer values to get the upper limit, lower limit, right limit, and the last one to get the left limit of each symbol in the image.

Algorithm (3.7): Char Segmentation

Input:

ProcessedLine[i][j]: array of line segmented image

H: High of the image,

W: Width of the image,

Nooflines: number of lines in line segmented image.

Output:

CharBorder : array of character segmented image.

Procedure:

```
1.  Set Si, Ei, noofchar =0
2.  For all x ∈ [0 , Noofline-1]
3.      Si=ProcessedLine[x].U
4.      Ei=PrecessedLine[x].D
5.      For all i ∈ [0 , 99]
6.          CharBorder[i].U= -1
7.          CharBorder[i].L= -1
8.          CharBorder[i].D= -1
9.          CharBorder[i].R= -1
10.     end For
11.     For all j ∈ [0 , W-1], i ∈ [$S_i$ , Ei]
12.         If (I[i][j] = 0) then
13.             If (CharBorder[Noofchar].L=-1)     Then
                    CharBorder[Noofchar].L=j
14.             Else  If (CharBorder[Noofchar].L≠-1)   Then
                        CharBorder[Noofchar].R=j,
                        CharBorder[Noofchar].U=Si,
                        CharBorder[Noofchar].D=Ei,
                Noofchar=Noofchar+1,
15.     end For
16.     Set R=0
17.     For all i ∈ [0 , Noofline-2], j ∈ [i+1 , Noofline-1]
```

```
            If (CharBorder[i].R = CharBorder[j].L-1)   Then
                CharBorder[i].R = CharBorder[j].R
                If (j=Noofchar-1)    Then   CharBorder[R].L=CharBorder[i].L,
                    CharBorder[R].R=CharBorder[i].R, R=R+1, i=j-1,
            Else  CharBorder[R].L=CharBorder[i].L,
                    CharBorder[R].R=CharBorder[i].R, R=R+1, i=j-1,
        end For
        Noofchar = R
end For
```

3.2.4 Further processing

After the character been separated, each character need more processing before being ready for the feature extraction step. In this step the character will be *shifted* into a standard position.

In shifting process, the character will simply be shifted towards the upper left corner of the image, so that all the characters will be almost at the same position of the image.

Algorithm (3.8): Shifting

Input:

CharImage[x][y]: The image that contains the character or digit extracted from the segmentation process,

CharHW: High and width of the image,

Output:

ShiftCharimage[x][y]: The image that contains the character or digit shifted towards the upper left corner of the image,

Procedure:

```
Set ShiftV=CharHW
For all $y \in [0 , CharHW-1]$, $x \in [0 , CharHW-1]$
        If ((CharImage[x][y] = 0) and (ShiftV>x)) then ShiftV=x
end For
Set u=o
For all $i \in [ShiftV-1 , CharHW-1]$, $j \in [0 , CharHW-1]$
        ShiftCharImage[u][j] = CharImage[i][j]
end For
u = u + 1
end For
```

3.2.5 Feature Extraction

Feature extraction process will generate feature sets that represent each character and digits in the training or testing image, in this work, features are extracted by using DCT (Discreet Cosine Transform) and two type of Wavelet transform (WT) Haar and Daubechies4.

3.2.5.1 Discrete Cosine Transform (DCT)

By using the DCT described in equation (2.5), each segmented image will be transformed from the original time domain to the frequency domain. The following algorithm was utilized to DCT transform the array of image data samples:

Algorithm (3.9): DCT Transform

Input:

A: is a block of two dimension array of pixels (the segmented symbol)

blocksize: the image size (height and width)

Output:

G: the transformed block of DCT coefficients

Procedure:

```
Set $C_1=1/sqrt(2)$,   Set $C_2=1/sqrt(2\times blocksize)$
For all $i \in [0 , blocksize-1]$, $j \in [0 , blocksize-1]$
    Set Sum=0
    For all $x \in [0 , blocksize-1]$, $y \in [0 , blocksize-1]$
        Set $w=((2\times y+1)\times j\times \pi)/(2\times blocksize)$
        Set $w2=((2\times x+1)\times i\times \pi)/(2\times blocksize)$
        Set $Sum=Sum+A[x,y]\times cos(w)\times cos(w2)$
    end For
    If (i=0) Then  $C_i=C_1$   Else  $C_i=1$
    If (j=0) Then  $C_j=C_1$   Else  $C_j=1$
    Set $G[i,j]=C_2\times C_i\times C_j\times Sum$
end For
```

And the inverse transform turns the data back to its original representation in the time domain (refer to equation (2.6)).

The output of the DCT is two-dimensional array to select a minimum number of coefficients and to convert it to one-dimensional array use the Zigzag ordering as shown in figure (3.7).

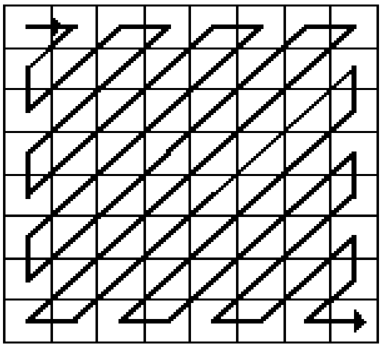


Figure (3.5): The Zigzag ordering

Algorithm (3.10): Zigzag ordering

Input:

DCT : is a two dimensional array contains the coefficients of DCT transform.

N : the length of block.

Output:

Zigzag: a one dimensional array contain the DCT coefficient

Procedure:

```
Set x, y, Right, Down =0, Set Zigzag[0]=DCT[x][y]
For all i ∈ [1 , N-1]
    If ((x=0)&(Right=0)) Then (Right=1, Down=0, y=y+1)
    Else If ((y=0)&(Down=0)) Then (Right=0, Down=1, x=x+1)
        Else If ((Right=1)&(Down=0)) Then (y=y-1, x=x+1)
            Else If ((Right=0)&(Down=1)) Then (x=x-1, y=y+1)
    Zigzag[i]=DCT[x][y]
end For
```

3.2.5.2 Wavelet Transform (WT)

Each character and digit image will be transformed by using wavelet transformation; coefficient produced in this transformation will act as features sets.

There are different types of wavelet transformations depending on the bases function used in the transformation, in our work; two type of wavelet transform are used to implement wavelet transformation these are Haar and Daubechies4 (Db4).

The transformation will be implemented by convolving the Haar and Db4 vectors with rows and then with columns, as explained in the previous chapter.

Wavelet will be applied to all characters and digits images have been extracted in the previous step; here is the algorithm of wavelet transformation used in this work [Umb98]:

Algorithm (3.11): Wavelet transformation

Input:

T[i]: The image of each character or digit image extracted from the segmentation process

N: Number of columns or rows.

Output:

W[i]: The transformed image of WT coefficients

Procedure:

- For each row in the image store row then apply **Haar WT** on the rows then store the new row in new image.
- For each column in the new image store column apply **Haar WT** on columns then store new column in last image.

Haar WT algorithm

```
1.    Set  L[0]= 1/sqrt(2), L[1]= 1/sqrt(2)
           H[0]= 1/sqrt(2), H[1]= -1/sqrt(2)
```

```
Set  Half = N/2
For all i ∈ [0 , Half-1]
    W[i] = T[2×i] × L[0] + T[2×i+1] × L[1]
    W[i+Half] = T[2×i] × H[0] + T[2×i+1] × H[1]
end For
For all i ∈ [0 , N-1]
    T[i] = W[i]
end For
```

It was be obvious that the Db4 WT algorithm could be applied with the same way of the Haar WT algorithm with only one different was the values of the low pass filter and high pass filter as mentioned in the previous chapter.

Wavelet features are computed for each sub-band using the following equation:

$$\text{Energy} = \sqrt{\sum_{i=iS}^{iE} \sum_{j=jS}^{jE} \left(W[i][j]\right)^2 \Big/ Size}$$

Where:
iS = is the start point on x-axes of sub-band,
iE = is the end point on x-axes of sub-band,
jS = is the start point on y-axes of sub-band,
jE = is the end point on y-axes of sub-band,
W[i][j] = the wavelet transform coefficients of the character or digit,
Size = (iE-iS+1) × (jE-jS+1).

3.2.6 Pattern Matching

In the pattern matching process the resulting test template, which is an N-dimensional feature vector, is compared against the stored reference templates to find the closest match. The process is to find how much unknown class matches a predefined class or classes. For the character recognition task, the unknown class is compared with all the predefined classes.

This comparison can be done through a *distance measure*, where the most common measure is the Euclidean Distance (E.D) which is the most common metric for measuring the distance between two vectors and is given by the square root of the sum of the squares for the difference between the two vectors component. Given the two vectors A and B, where A = [a_1 a_2 … a_N] and B= [b_1 b_2 … b_N], then the Euclidean distance is given by [Umb98]:

$$E.D. = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2}$$

3.2.7 Decision Rule

The decision rule process is to select the pattern that best matches the unknown sample. For the proposed character recognition system, the minimum distance between two features is considered to assign the unknown pattern to the nearest predefined pattern. That is, the identity of the unknown character is recognized as the best matched reference in the database.

3.2.8 Database References

Database is made to support the training and testing processes of the system. The proposed character recognition system depends on the samples of characters (capital & small letters) and digits (0 to 9) extracted in the training phase. This database contains 26 capital letters, 26 small letters, and the 10 digits, so the total number of database sample is:

$$\begin{aligned} \textit{Total DB Sample} &= \textit{capital letters} + \textit{small letters} + \textit{10 digits} \\ &= 26 + 26 + 10 \\ &= \textit{62 sample} \end{aligned}$$

3.3 The Definition of Recognition Ratio

The recognition rate is defined as the ratio of correct identified characters to the total number of database sample in the test set corresponding to a nearest neighbor decision rule.

$$\textit{Recognition Rate} = \frac{\text{Correct Recognized Character}}{\text{Total number of Tested Characters}} \times 100\,\%$$

Chapter Two

Theoretical Background

2.1 Introduction

Many attempts have been made to make text recognition reaches a level of performance that is close to emulating human capabilities in performing general image analysis function. Research in biological and computational systems continually is uncovering new and promising theories to explain the human visual cognition, however, the state of art in computerized image analysis for the most part is based on heuristic formulation tailored to solve specific problems, for examples some machines are capable of reading printed, properly formatted documents at a speed that are orders of magnitude faster than the speed that the most skilled human reader could achieve. However the systems of this type are highly specialized and thus have a little or on extendibility. That is, current theoretic and implementation limitations in the field of image analysis imply solution that are highly problem dependent [Gon92].

This chapter concerned with the pattern recognition systems that will be applied to the proposed character recognition system, describing images formats, text recognition system phases, operation required within each phases and some feature extraction techniques.

2.2 Image File Formats

The study of text recognition or in general pattern recognition required some knowledge on image processing. The first step is to have knowledge in how to store image file. There are many different types of images and applications with varying requirements and also considers market share, proprietary information, and a lack of coordination within the imaging industry. However, some standard file formats have been developed, and the ones presented here are widely available. Many other image types can be readily converted to one of the types presented here by easily available image conversion software.

Some of the images are compressed of different compression quality and others are kept without compressed because high quality is required. There is a BMP (BITMAP), TIFF (Tagged Image File Format), GIF (Graphics Interchange Format), JPEG (Joint Photographic Expert Group), IMG (IMAGE)…and other image files, where each of them has its specific format. Each pixel in the digital BMP image is representing by 24, 8, 4, or 2 bits. In this research an 8 bits BMP image is used for reason of its standard use.

2.3 Pattern Recognition

The pattern recognition is a task of finding some conceptual and relevant information from raw data. The application used here is Text recognition, and the raw data used is image of characters and digits.

2.4 Text Recognition Systems

In general any text recognition system consists of two phases:

Building DB Phase and Testing Phase

- Building DB Phase: It is the phase that made the preprocessing on the input training image to remove noise, then segments the training input image into meaningful segments each segment contains a character or digit and built the database of each symbol in the input image with the feature extraction techniques used in the system.

- Testing Phase: It is the phase that made the preprocessing on the input testing image to remove noise, after that, segments the image into meaningful segments each segment contains a character or digit, then, extract the features extraction for each segment in the input image with the feature extraction techniques used in the system, finally, a pattern matching process will be applied on each testing symbol to find how much it matches a predefined training symbols in the database to extract the most similar character.

Techniques used in each phase could be divided into three basic levels; figure (2.1) illustrates these levels [Gon92]:

(1) Low level processing,

(2) Intermediate processing,

(3) High level processing.

Although these subdivisions have no definitive boundaries, they do provide a framework for categorizing various processing that is inherent component of an autonomous pattern recognition system.

2.4.1 Low Level Processing

This level deals with function that may be viewed as automatic reactions, requiring no intelligence on the part of the image recognition system. Image acquisition and preprocessing are considered here as low level function; this area encompasses activities from the image formation process itself to compensation, such as noise reductions and image deblurring. Low-level function may be compared to the sensing and adapting processes that we can not begin until a suitable image is available. The process followed by the brain in adapting the visual system to produce such an image is an automatic unconscious reaction.

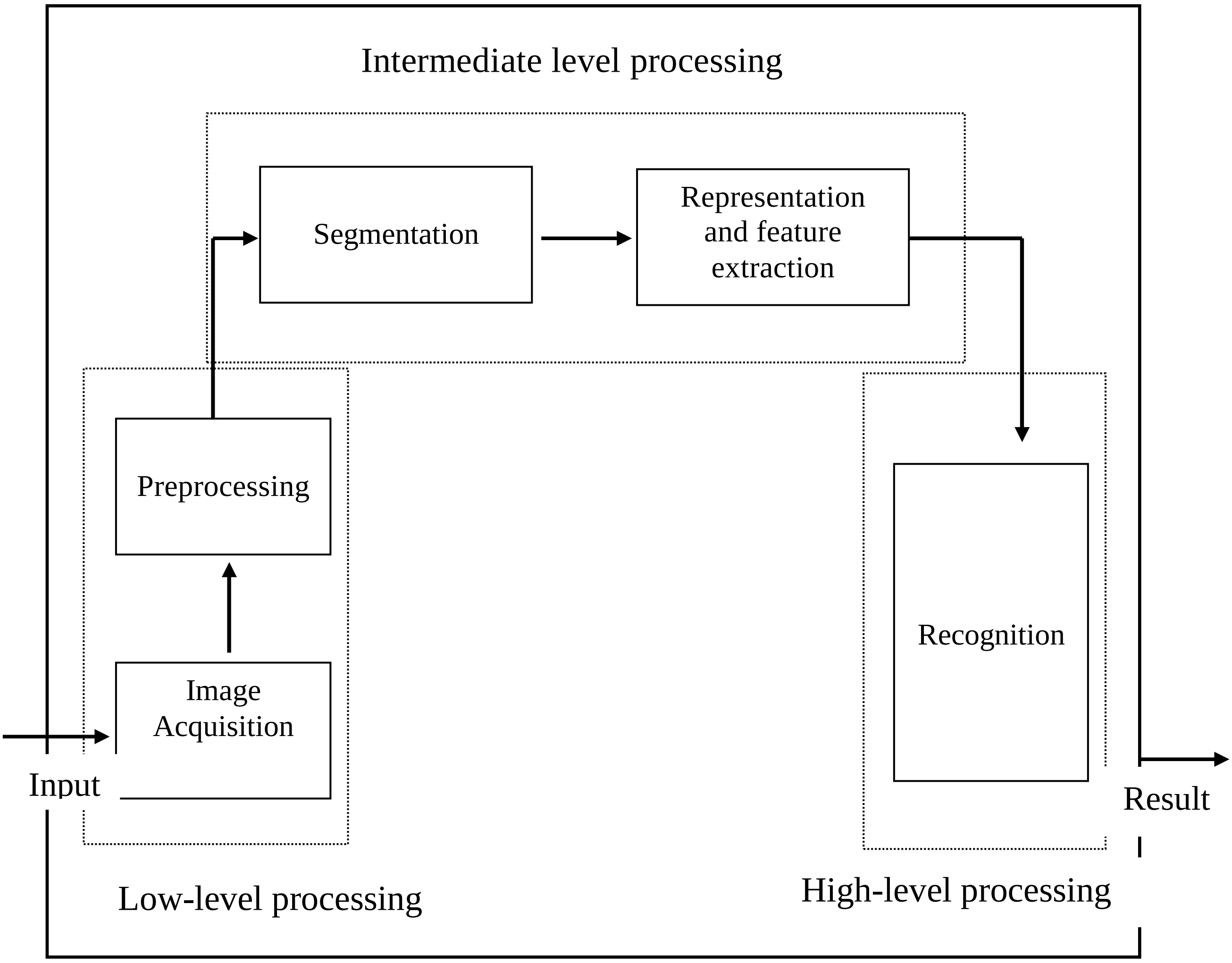


Figure (2.1): Text Recognition System

2.4.1.1 Image Acquisition

This process is to acquire digital image from physical world, by using scanner or digital camera. The image produced by the imaging device is stored as uncompressed BMP gray scale image contain the data that is to be processed, and further information will be extracted from it, therefore, image status, like resolution, brightness, and defected comes from low performance of the imaging device will highly affects the next processing steps and as consequence will define limits to the performance of the recognition system and the recognition rate [Umb98].

2.4.1.2 Preprocessing

Improve the input image character coordinate sequence data, involves several activities, such as smoothing, and noise removing. This step yield two benefits, the reduction of noise and the reduction of time usage in other steps [Kur97].

The key function of preprocessing is to improve the image in a way that increases the chances for success of the other processes, for character recognition, the preprocessing is usually deals with techniques of image enhancement, noise removing, binarization, and other techniques required such as shifting and rotation. Rotation is a tool to rotate an image about its center by the specified number of degrees. The input image entered to the system using scanner, so it could be rotated by fault of user. Because rotation requires special techniques (especially for the decision of rotation angle) this research did not deal with rotation.

a. Histogram Stretching

Histogram stretching is the technique that used to reduce the difference between gray level values in the image which could be known as the accuracy of each gray level value. This method selects and stretches the main power interval of the histogram. The calculation of the power interval is selected by a minimal and a maximal point. These points are determinate by a threshold value. After that, the main power range is stretched to the full range (0-255) [Sta98], the following mapping function is utilized to compute pixel brightness values [Kur97]**:**

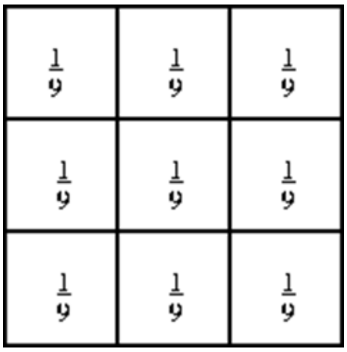
$$\text{Output}[x][y] = ((\text{Input}[x][y]) - I_{MIN}) / (I_{MAX} - I_{MIN}) \times 255) \quad \text{……..} \quad (2.1)$$

Where:
x: is the height of the image,
y: is the width of the image,
I_{MIN}: is the minimum value of the input image,
I_{MAX}: is the maximum value of the input image,

b. Noise Removing

Noise removing is one of the image enhancement application in which the principle objective is to process an image so that the result is more suitable than the original for specific application. Smoothing filters are used for giving an image a softer effect or to eliminate noise like mean, median and Gaussian filters [Gon00]. In this research a mean filter is used cause of it removes noise without scratching the symbols on the text image.

This filter is low pass filter, it smoothen the image to much the pixels nearby in a way that no point in the image differ from its surroundings to a greater extent. Mean filtering is a simple and easy to implement method of smoothing images; it is often used to reduce noise in images. The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. There are different sizes for this filter i.e. 3×3, 5×5, and 7×7. But, a 3×3 filter size is used in this research to avoid losing the shape of symbols in the image, as shown in figure (2.2).



$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figure (2.2) 3×3 mean filter

c. Binarization

It is the process of turning a gray scale into a binary image (only two levels of interest 0 and 1) in order to improve the contrast between the text (isolated characters) and the back ground of the image and consequently facilitates the feature extraction process.

To perform binarization for an image, a threshold value in the gray scale image is picked. Every darker (lower in value) than this threshold value is converted to black and every thing lighter (higher in value) is converted to white. The black pixels (of value 0) represent a pixel of character, and the white pixels (of value 1) represent the back ground of the image [Blo03].

The difficulty with binarization lies in finding the right threshold value to be able to remove unimportant information and enhance the important one. It is impossible to find the working global threshold value that can be used on every image. The variation can be too large in these types of text images. Therefore algorithms to find the optimal value must be applied separate on each image to get a functional binarization. There are a number of algorithms to perform this, the simplest one use the mean value or the median of the pixel values of the image [Blo03].

All of these algorithms are based on global thresholds. What are often used nowadays are the localized thresholds. The image is partitioned into smaller blocks and threshold value is then calculated for each of those blocks as threshold value for their specific block. Local threshold demand an increase calculation time, but mostly compensate it with a better result.

The global mean threshold for a block of size H×W is defined as:

$$GlobalMean = \frac{1}{H * W}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} image(i, j) \ldots (2.2)$$

Where:

image(i,j): is the gray level at pixel (i.j) of the image.

The local mean threshold for a block of size W×W is defined as:

$$LocalMean(k) = \frac{1}{W^2}\sum_{i=0}^{W-1}\sum_{j=0}^{W-1} Block(i, j) \ldots (2.3)$$

Where:

LocalMean(k): is the computed mean gray level value for k^{th} block,

Block(i,j): is the gray level value at pixel (i,j).

The local mean threshold, make it possible to find different threshold values for each part of the text image depending on the level of darkness of each part. For that reason the local mean threshold was adopted in this research.

2.4.2 Intermediate Level Processing

In this level of processing, techniques are made to characterize image component and extract information needed for recognition. Segmentation techniques is required to segment the text into separated characters, and the feature extracting techniques is required here in way that each character will be described by a set of features rather than it’s raw representation, at which all characters will be recognized based on those features.

2.4.2.1 Segmentation

Partitioning of an image into several constituent components is called segmentation. Segmentation is an important part of practically any automated image recognition system, because it is at this moment that one extracts the interesting objects, for further processing such as description or recognition. Division of the image into regions corresponding to objects of interest is necessary before any processing can be done at a high level processing. Segmentation is one of the important elements in automated image analysis because it is at this step that object or other entities of interest are extracted from an image for subsequent processing, such as description and recognition. For example, if the goal is to recognize black characters, on a white background, pixels can be classified as belonging to the background or as belonging to the characters: the image is composed of regions which are in only two distinct dark text and white background.

The segmentation process in this research is taking two steps the line segmentation step and the character segmentation step; they will be explained in details in the third chapter (practical part).

2.4.2.2 Feature Extraction

Object recognition is generally performed on the raw image in the image plane or on the feature representation in the feature space. In the earlier case, known the low-level image recognition, the system learns and recognizes an object according to the information given by all the pixels in the image plane. In an N×N image plane, the object is described by an image vector, which consists of N^2 pixel values. The size of the image vector increases as the resolution of the object image increases [Cha95].

One of the drawbacks of this approach is a huge dimensionality, which deepens the computational burden of the system. Moreover, the image vector of the shifted object image may be quite different from the original one. On the other hand, not all the pixels of the object image reveal crucial information of the object characteristics, and there is a large redundancy in the image vector [Cha95].

The feature-based recognition uses only the information that best characterize the object. It extracts the important information conveyed by some pixels and processes it to obtain the feature representation. The object in the image plane is then represented by its feature vector in the feature space. In this case, the learning and the recognition is done in the feature space. Dimensionality of the input vector is greatly reduced, and the recognition can be invariant to some image transformations, such as image translation, rotation, and scaling, if the object features are properly selected [Mac96].

Many methods have been introduced to extract features; generally it could be classified into two main categories:

1. Geometrical features: those features that are directly extracted from the image (spatial domain). Many techniques have been introduced in these categories that describe the shape of the pattern.
2. Transformation features: those features that are extracted after a transformation is made up to map the image from the spatial domain into another domain.

Feature transformation is a process through which a new set of features is created which is used to define each symbol (character or digit) segmented from the text image by getting its features, so it is used in this research.

2.4.3 High Level Processing

This level involves recognition and interpretation; these two processing have a strong resemblance to what generally meant by intelligent cognition. The majority of techniques used for low and intermediate processing encompass a reasonably well-defined set of theoretic formulations. However, as we venture into recognition, and especially into interpretation, our knowledge and understanding of fundamental principles becomes far less precise and much more speculative. This relative lack of understanding ultimately results in a formulation of constraints and idealization intended to reduce task complexity to a manageable level. The end product is a system with highly specialized operational capabilities [Gon00].

The recognition could be in different ways like:

1. Classification
2. Pattern matching (Euclidean Distance, City Block)

1. Classification

In general term, classification is the process of assigning each object, from set of objects, to one set of classes, in pattern recognition, the object is a pattern extracted from the image, and the classes are various categories occurring in the image. The pattern in this step of processing not usually a set of points, but, it is a set of numerical features formed by feature extraction process [Nib86].

There are different models of classification system; the main three modules are statistical, syntactical, and artificial neural network. Since most of neural networks approaches are based on statistical method [Ven93]. Statistical Classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc) and based on a training set of previously labeled items. And Neural network can be defined as information processing systems that have certain performance characteristics in common with biological neural network that actually the basic principle of neural network working is abstract simulation of real neuron system [Fu94].

2. Pattern matching

In the pattern matching process the resulting test template, which is an N-dimensional feature vector, is compared against the stored reference templates to find the closest match. The process is to find how much unknown class matches a predefined class or classes. For the character recognition task, the unknown symbol is compared with all the predefined symbols. This comparison can be done through a distance measure, where the most common measure is Euclidean Distance and City Block [Mus06].

a. Euclidean Distance (E.D.)

It is the most common metric for measuring the distance between two vectors and is given by the square root of the sum of the squares for the difference between the two vectors component. Given the two vectors A and B, where A = [a_1 a_2 … a_N] and B= [b_1 b_2 … b_N], then the Euclidean distance is given by [Umb98]:

$$E.D. = \sqrt{\sum_{i=1}^{N} (a_i - b_i)^2} \quad \text{..........} (2.4)$$

b. City-Block Distance (C.D.)

Another distance measure, called the City-Block distance or absolute value metric, is defined as follows (using A and B vectors as before) [Umb98]:

$$C.D. = \sum_{i=1}^{N} |a_i - b_i| \quad \text{...........} (2.5)$$

Since in this research the aim was only to classify the printed characters (with given font and size), the pattern matching techniques are adequate to implement the classification.

Transformation Methods

Transformation methods are used to map the signal from one domain representation to another (e.g. from the spatial domain to the frequency domain), these transformation methods are:

2.5.1 The Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a technique for converting a signal into elementary frequency components. The image is transformed from spatial domain to frequency domain using two dimensions (2D) DCT basis function. The DCT formula is given by: [Sal00]

$$G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p_{xy} \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \quad \text{.......... (2.6)}$$

Where:

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0. \\ 1, & f > 0. \end{cases}$$

i,j: 0…N-1

N: the height or width of the image.

To turn the image back to its original domain the inverse transform must be applied to the transformed signal, the Inverse DCT is given by:

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_i C_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \quad \text{...... (2.7)}$$

Where

P: is the original image,

x,y: 0,…N-1

G: is the transformed image,

N: the height or width of the image.

The DCT frequency coefficient is described in figure (2.3), the element in the upper-left corner is the low frequency coefficient for the entire two-dimensional array, and all the remaining coefficients contain high frequency information [Web2].

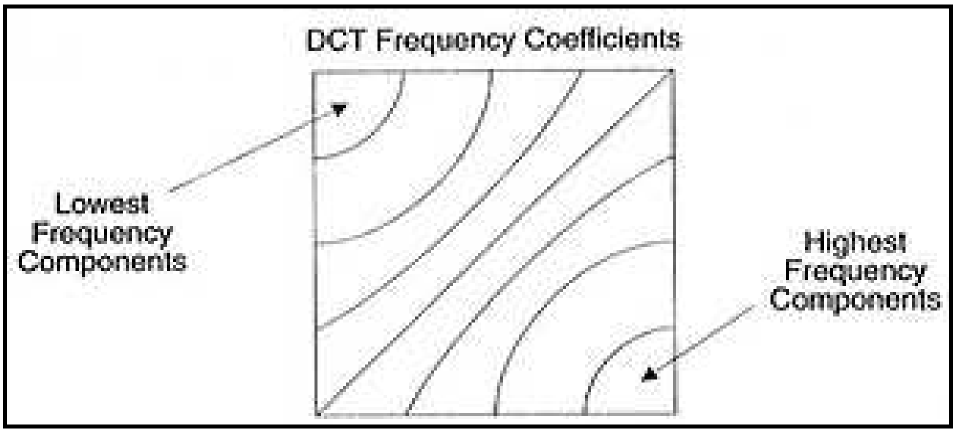

DCT Frequency Coefficients
Lowest Frequency Components
Highest Frequency Components

Figure (2.3): The DCT Frequency Coefficients

2.5.2 The Wavelet Transform (WT)

Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale. Wavelets have scale aspects and time aspects. They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology. Interchanged between these fields have led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction [Gra95].

A wavelet is a waveform of effectively limited duration that has an average value of zero [Mat00]:

$$\int_{-1}^{+1} \varphi(t)\,dt = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2.8)}$$

This is dilated with a scale parameter **s**, and translated by **u**:

$$\varphi_{us}(t) = \frac{1}{\sqrt{s}}\varphi\left(\frac{t-u}{s}\right) \text{........................} (2.9)$$

The wavelet transform of ***f*** at the scale **s** and position **u** is computed by

$$Wf\ (u\,,\ s) = \int_{-1}^{+1} f(t)\frac{1}{\sqrt{s}}\varphi\left(\frac{t-u}{s}\right)dt \text{ ....} (2.10)$$

2.5.2.1 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is a special case of the WT that provides a compact representation of a signal in time and frequency that can be computed efficiently. The DWT is defined by the following equation:

$$DWT\ (s,u) = \sum_{1}^{s}\sum_{1}^{u} f(u)2^{-s/u}\varphi\left(2^{-s}n-u\right) \text{........} (2.11)$$

Where:

φ(t): is a time function with finite energy called mother wavelet.

DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and high pass filters, respectively.

The decomposition of the signal into different Frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal.

The original signal **x[n]** is first passed through a half band high pass filter **g[n]** and a low pass filter **h[n]**. After the filtering, half of the samples can be eliminated. The signal can therefore be sub sampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and mathematically is expressed as follows:

$$Y_{high}[k] = \sum_{n} x[n].\,g[2k - n] \text{ .... } (2.12)$$

$$Y_{low}[k] = \sum_{n} x[n].\,h[2k - n] \text{ .... } (2.13)$$

Where: $Y_{high}[k]$, $Y_{low}[k]$ are the outputs of the high-pass (g) and low-pass (h) filters.

This process is further shown in figure (2.4) [Jen01].

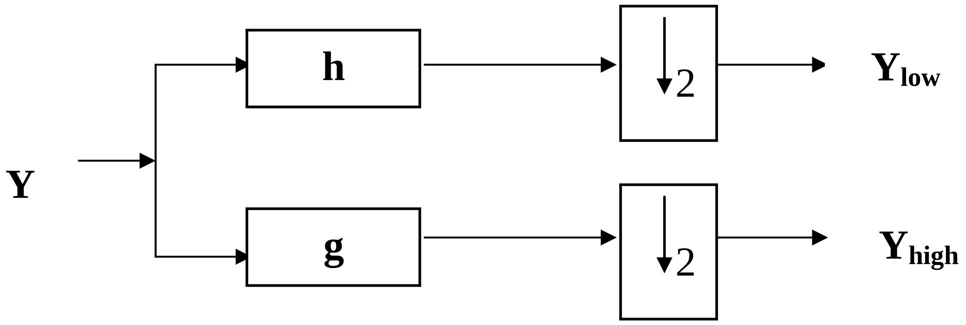


Figure (2.4): High-pass and low-pass Filters in DWT

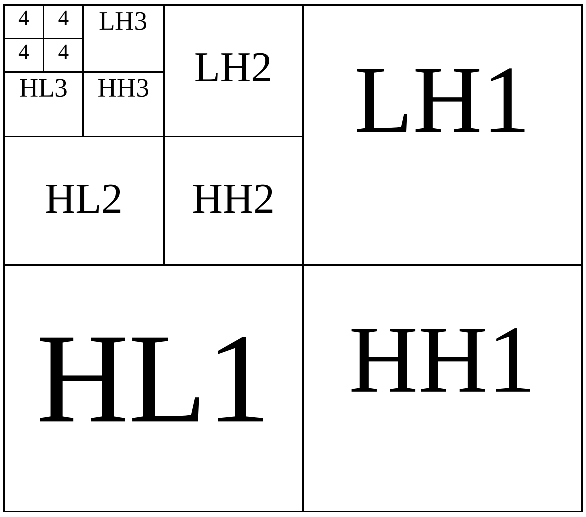


Figure (2.5): WT decomposition of an image for four resolution levels

The DWT transform is subband transform. It is done by computing a convolution of the data with a set of bandpass filters. Each resulting subband encodes a particular portion of the frequency content of the data. The principle of DWT transform is to calculate averages and difference. It partitions the image into regions such as that one region contains large numbers (averages), and the other regions contain small numbers (differences). However, these regions, which are called subbands, are more than just sets of large and small numbers.

There are two type of wavelet used here Haar transform and Daubechies4 transform, they work at the same way with only one difference their base vectores:

The Haar base vectors are: [Kap02]

Low-pass filter: $\frac{1}{\sqrt{2}}[1\ ,1\ ]$

High-pass filter: $\frac{1}{\sqrt{2}}[1\ ,-\ 1\ ]$

While the Db4 base vectors are: [Kap02]

Low-pass filter: $\left[\frac{1+\sqrt{3}}{4\sqrt{2}},\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{3-\sqrt{3}}{4\sqrt{2}},\frac{1-\sqrt{3}}{4\sqrt{2}}\right]$

High-pass filter: $\left[\frac{1-\sqrt{3}}{4\sqrt{2}},\frac{-3+\sqrt{3}}{4\sqrt{2}},\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{-1-\sqrt{3}}{4\sqrt{2}}\right]$

The transformation will be implemented by convolving these vectors with rows and then with columns.

One sub-band has been high-pass filtered in both horizontal and vertical directions, one has been high-pass filtered in vertical direction and low-pass filtered in horizontal direction, one has been low-pass filtered in vertical direction and high-pass filtered in horizontal direction, and one that has been low-pass filtered in both directions. LL_1 will corresponds to the lowest frequency and contains the global characteristics of the image, LH_1 will gives the vertical high frequencies and contains the horizontal details, HL_1 will give the horizontal high frequencies and contains the vertical details, and HH_1 will give the high frequencies in both diagonal directions and contains the diagonal details, as shown in figure (2.5) [Gon00].

الخلاصة

أن تمييز الحرف يلعب دوراً مهماً وكبيراً في العالم الحديث. حيث يعتبر تمييز الحرف حلاً مهماً للعديد من المشكلات لإستخدامه في العديد من المجالات و التطبيقات في مجال تمييز الانماط. أن المشكله الحقيقيه هي في تصميم نظام يستطيع استخلاص الخواص الصحيحه بكفاءه. لذلك فأن اغلب الاهتمام ينصب على ايجاد سلسله من المعالجات المسبقه، وخوارزميه تقطيع الحروف وبالتالي الطرق الكفؤه في استخلاص الخواص.

في هذا البحث تم استخدام عمليات مسبقه مهمه على الصوره الداخله و ذلك بهدف زيادة معدلات تمييز الاحرف. حيت تم استخدام عدد من الخطوات في المعالجه المسبقه و تشمل Histogram stretching، Mean Filter و أخيرا عملية Binarization. ثم يُتبع ذلك عمليه التقطيع وتحديد حدود الحرف. كما تم استخدام عدد من عمايات استخلاص الصفات في هذا البحث مثل (DCT) (تحويل دالة الجيب تمام المحددة) وكذلك (HWT و Db4) (انواع من التحويل المويجي).

إن عملية توافق ألانماط مع قاعدة البيانات المخزونه للرموز المستخدمه تتم بأستخدام Euclidean Distance ولقد تم حساب معدلات التمييز كمقياس لكفاءة كل طريقه و تبين نيجه البحث أن استخدام ال (DCT) تعطي افضل نتائج عندما يتم استخدام ٢٥ عاملا او اكثر في استخلاص الخواص. مع ملاحظة ان زيادة عدد العوامل يؤدي الى زيادة وقت المعالجه.
اما بالنسبه لطريقه WT بنوعيها Haar, Db4، فان معدلات التمييز فيها تكون أقل من الطريقه الاولى. مع ملاحظة ان هذه الاساليب قد تم فحصها باستخدام عدد قليل نسبيا من العوامل.

Abstract

Character recognition plays an important role in the modern world. Character recognition is considered to be very important solution to many problems because of its numerous applications and theoretical values in the domain of pattern recognition. The only real problem is to design a system that can extract the right features efficiently. Thus, most of the attention is devoted to find the optimal sequence of preprocessing techniques, character segmentation procedure, and efficient feature extraction methods.

In this thesis important operations are used in preprocessing of the input image in a way that increases the character recognition rates. A number of steps are used in the work namely: Histogram stretching, Mean Mask and finally Binarization technique. This is followed by segmentation process and fixing character boundaries. Number of feature extraction techniques has been used to obtain the features. These are Discreet Cosine Transform (DCT), and two Wavelet Transform (WT) techniques (Haar and Daubeche-4 or Db4).

Pattern matching with database for letters and decimal digits using Euclidean Distance as decision metric. Recognition rates are computed and used as performance measure for each method. The results have shown that the use of DCT method gives the best results when 25 or more coefficients are involved in the feature extraction. Using larger number of coefficients require more processing time. On the other hand wavelet based techniques provide slightly less recognition rates. These techniques are tested with very small number of coefficients.

Table of Contents

Contents **Page**

References

- **[Awc95]** G. J. Awcock and Thomas; “Applied Image Processing”, Macmillan New Electronics Series, 1995.

- **[Blo03]** J. Blomme, “Evaluation of biometric security systems against artificial fingers”, Linkoping University, LITH-IS-EX-3514, October 2003. http://www.ep.liu.se/exjobb/isy/2003/3514/

- **[Bon97]** K. S. Bong, "Ligature Modeling for On-line Cursive Script Recognition", IEEE Trans. Pattern Recognition and Machine Intelligent, 19 (6), June 1997.

- **[Bro92]** E. W. Brown**,** "Applying Neural Networks to Character Recognition", Northeastern University internal paper, 1992. http://www.ccs.neu.edu/home/feneric/Papers/charrecnn.pdf

- **[Cha03]** B. B. Chaudhuri, "A Majority Voting Scheme for Multiresolution Recognition of Handprinted Numerals", 2003, www.csc.liv.ac.uk/ ~prima/ICDAR2003/Papers/0002_620_bhattacharya _u.pdf

- **[Cha95]** S. K. Chai, "Multi Classifier Neural Network for handwritten Character Recognition", Ph.D. Thesis, Department of Computer Science, University of Toronto, Canada, 1995.
- **[Cov00]** Covavisaruch Nongluk, Apichart Sajjapong1, and Wiwat Vatanawood, "On-line Handwritten Thai Characters Recognition Systems", Vol, No. 133, 2000. http://cgci.cp.eng.chula.ac.th/Publications.html,2000.

- **[Ema95]** Emad Jassim Mohammed, "Arabic Character Recognition Methodology using Complex Moments", M. Sc. Thesis, Department of Computer Science, University of Technology, Iraq, 1995.

- **[Fu94]** Fu L, "Neural Network in Computer Intelligence", University of Florida, McGraw Hill, Inc, 1994.

- **[Gon00]** Gonzalase R. C., Woods R. E., "Digital Image Processing", Addison-Welsley publication Company, Inc., 2000.

- **[Gon92]** Gonzalase R.C., "Digital Image Processing", Addison-Welsley publishing Company, 1987.

- **[Gra95]** A. Graps, "An Introduction to Wavelets", IEEE Computational and Engineering, 1995, http://www.amara.com/current/wavelet.html.

- **[Imp91]** S. Impedove, L. Ottaviano, and S. Occhinegro, “Optical character recognition – a survey,” Internal Journal of Pattern Recognition and Artificial Intelligence, Vol. 5 (1-2), pp.1-24, 1991.

- **[Jen01]** Jensen, A. and Harbo, A., C.; "Ripples in Mathematics - The Discrete Wavelet Transform", Springer, Berlin, 2001

- **[Kap02]** Kaplan, I.; "A Linear Algebra View of the Wavelet Transform",January-2002. www.bearcave.com/misl/misl_tech/wavelets/daubechies

- **[Kur97]** Kurz, Ludwik & Hafed, M. "Analysis of Variance in Statistical Image Processing" Cambridge University Press, 1997.

- **[Lee96]** Lee, S.W. "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network". IEEE Trans. On Pattern Analysis and Machine Intelligence, 18(6):648-652, 1996.

- **[Mac96]** R. B. Macy, "Pattern Recognition with Neural Network in C++", IEE Press, 1996.

- **[Mal02]** S. M. Maliki, "Character Recognition Using Neural Network", M. Sc. Thesis, Department of Computer Science, Baghdad University, Iraq, April 2002.

- **[Mar94]** P. Mark and P. Rejean, " Machine vs. Human in a Cursive Script Reading Experiment Without Linguistic Knowledge", Proc. 12th ICPR, Jerusalem, Vol. 11, 1994, pp.93-98.

- **[Mat00] MatLab** Documentation, "Wavelet Technique", 2000. http://www.mathworks.com/support/product/seemore.html?page=1&type=Documentation&product=ML

- **[Mus06]** Mustafa D. Al-Hassani, "Identification Techniques Using Speech Signals and Fingerprints", P. hD. Thesis, Department of Computer Science, Al-Nahrain University, Iraq, March 2006.

- **[Nib86]** Niblack W, "An Introduction to Image Processing", Prentice Hall, UK, 1986.

- **[Nik98]** Nikola K. Kkasabov "Foundations of Neural Networks Fuzzy Systems, and Knowledge Engineering", Mssuchsettes Institute of Technology, 1998.

- **[Par99]** Park Jaehwa; “hierachical character Recognition and it’s Use in Handwritten Word/Phrase Recognition”, Ph. D. thesis, University of New York at Buffalo, 1999.

- **[Pav92]** T. Pavdilis and Z. Jiangying, “Page segmentation and classification,” CVGIP: Graphical models and image processing, Vol.54, No.6, 1992.

- **[Pla02]** R. Plamondon and S. N. Srihari, “On-line and off-line handwriting recognition: a comprehensive survey,” IEEE Transactions on Pattern Analysis and Machine Vision, Vol.22, No.1, January 2002.
- [**Sal00]** D.Salomon, "Data Compression the Complete Reference", Addition Wesley Company, Second Edition, 2000.

- **[Ser03]** Sergios Theodoridis, "PATTERN RECOGNITION SECOND EDITION", Department of Informatics and Telecommunications University of Athens Greece, 2003.

- **[Sta98]** J. L. Starck, F. Murtagh, & A. Bijaoui, "Image Processing and Data Analysis: The Multiscale Approach", Cambridge University Press, 1998.

- **[Tap90]** C. Tappert, et al; “The State of the Art in On-line Handwritten”, IEEE Trans. Pattern Recognition and Machine Intelligent, 12 (8), June 1990, pp. 787-808.

- **[Umb98]** Scott E. Umbaugh, "Computer Vision and Image Processing: A practical Approach using CVIP tools", (Ph.D.) Thesis, Prentice-Hall PTR. Inc., 1998.

- **[Vam04]** K. M. Vamsi, "printed Character Recognition using Neural Networks", School of ITEE, University of Queensland, Australia, 2004.

- **[Ven93]** A. N. Venetsanopoulas, "Artificial Neural Network Learning algorithms Performance, Evolution, and Application", Kluwer Academic Publishers, Massochusetts, USA, 1993.

- **[Yan06]** H. Yan, "Structural boundary feature extraction for printed character recognition", Department of Electrical Engineering, The University of Sydney, Volume 1451, 2006 NSW, Australia

[Web Sites]

- **[Web1]** “Code: a system in which arbitrary values are given to letters, words, numbers or symbols to ensure secrecy or brevity,” http://homepages.cwi.nl/~dik/english/codes/charrec.html.

- **[Web2]**"What is Discrete Cosine Transform (DCT)", http://www.fh-friedberg.de/fachbereiche/e2/telekom-labor/zinke/mk/mpeg2beg/ whatisit.htm

Appendix A

BMP Image-File format

BMP Image-File Format [Sama99]

The BMP file format divides a graphics file into four major parts, these are:

- Bitmap File Header: The bitmap File Header is 14-byte long and is formatted as follows:

UNIT	bfType	*holds the signature value 0x4D42, which identify the file as BMP*
DWORD	bfSize	*holds the file size*
UNIT	bfReserved	*not used, set to zero*
UNIT	bfReserved	*not used, set to zero*
DWORD	bfOffBits	*specifies the offset, relative to the beginning of the file, where the data representing the bitmap itself begins*

- Bitmap Information Header: The bitmap information header is 40-bytes long and it contains important information about the image. The windows format for this header is:

DWORD	biSize	*holds the header length in bytes*
LONG	biWidth	*Identify the image width*
LONG	biHeight	*Identify the image height*
WORD	Bitplanes	
WORD	biBitCount	*Identify number of bits/pixel in the image and thus the maximum number of colors that the bitmap can contain*
DWORD	biCompression	*Identify the compression scheme that the bitmap employs. It will contain zero if the bitmap uncompressed*

DWORD	biSizeImage	*Set to zero for uncompressed image, else it holds the size (in bytes) of the bits representing the bitmap image for compressed image*
LONG	biXPelsPerMeter	
LONG	biXPelsPerMeter	
DWORD	biClrUsed	
DWORD	biClrImportant	

- Palette (Color table containing RGB quad or RGB triple structure): the color table specifies the colors used in the bitmap. The BMP files come in four color formats:

1.	2-color	*1-bits per pixel*
2.	16-color	*4-bits per pixel*
3.	256-color	*8-bits per pixel*
4.	16.7 million-color	*24-bits per pixel*

The number of bits per pixel -and hence the color format-can be determined from the biBitCount shown above.

In the 2-color, 16-color, and 256-color BMP formats, the color table contains one entry for each color. Each entry specifies the intensities of a color's *red, green, and blue* component and it is of 4-bytes long as shown below:

BYTE	rebBlue
BYTE	rebGreen
BYTE	tebRed
BYTE	rebReserved

Each color-table entry can specify a range of red, green, and blue values from 0 to 255. true-color BMP files do not contain color tables, because a single color table with 16.7 million entries of 4-bytes each would require 64MB of storage space.

- Bitmap Bits: The bitmap bits are the set of bits defining the image-the bitmap itself. In the 2-color, 16-color, and 256-color BMP formats, each entry in the bitmap is an index to the color table. In a 16.7 million-color bitmap, where there is no color table, each bitmap entry directly specifies a color. The first 3-bytes in each 24-bit entry specify the pixel colors red component, the second specifies green component and the third specifies blue.

  The bitmap bits representing a single line are stored in left-to-right order, the same way that the pixels they represent line up on the screen. The first row pixel data in the bitmap corresponds to the bottom row of pixel on the screen, the second row corresponds to the row of pixels second from the bottom, and so on.

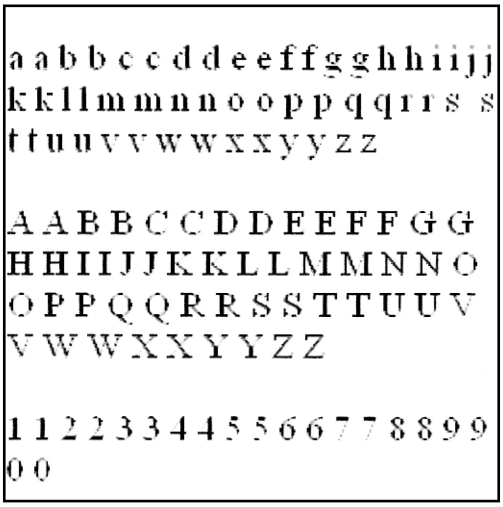
  The size of one bitmap entry is determined by the number of bits per pixel assigned to each color format, as shown in the following table:

Number of colors	Number of bits per pixels required
2	1
16	4 (1/2 byte)
256	8 (1 byte)
16.7million	24 (6 bytes)

Appendix B

The Input Images

This section is devoted for displaying the ten input tested images. These are used as inputs to our proposed character recognition system model. These images are scanned with 200 dpi and 256 gray-scales with BMP format.

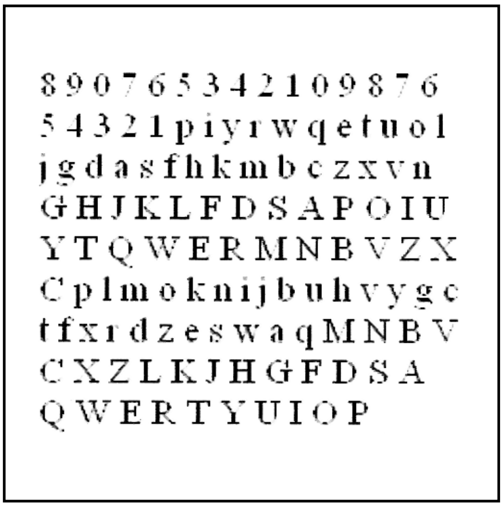


a a b b c c d d e e f f g g h h i i j j
k k l l m m n n o o p p q q r r s s
t t u u v v w w x x y y z z

A A B B C C D D E E F F G G
H H I I J J K K L L M M N N O
O P P Q Q R R S S T T U U V
V W W X X Y Y Z Z

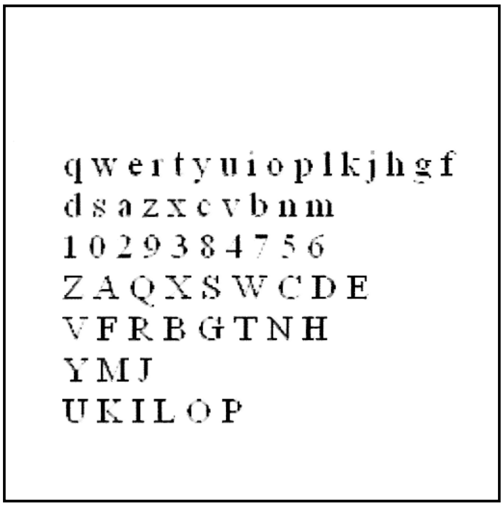
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9
0 0

Text 1



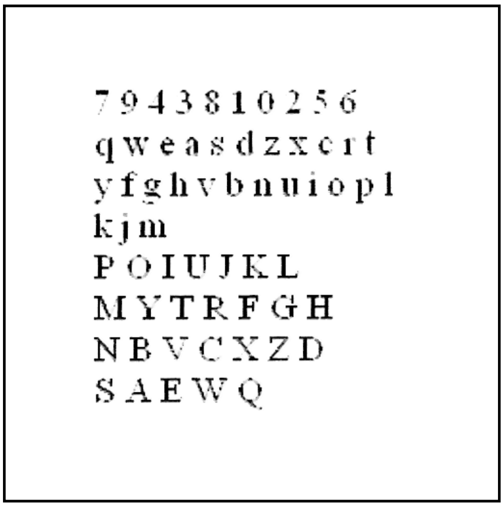
8 9 0 7 6 5 3 4 2 1 0 9 8 7 6
5 4 3 2 1 p i y r w q e t u o l
j g d a s f h k m b c z x v n
G H J K L F D S A P O I U
Y T Q W E R M N B V Z X
C p l m o k n i j b u h v y g c
t f x r d z e s w a q M N B V
C X Z L K J H G F D S A
Q W E R T Y U I O P

Text 2



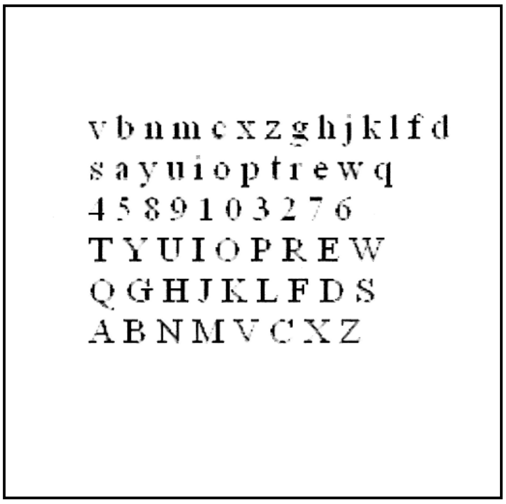
q w e r t y u i o p l k j h g f
d s a z x c v b n m
1 0 2 9 3 8 4 7 5 6
Z A Q X S W C D E
V F R B G T N H
Y M J
U K I L O P

Text 3



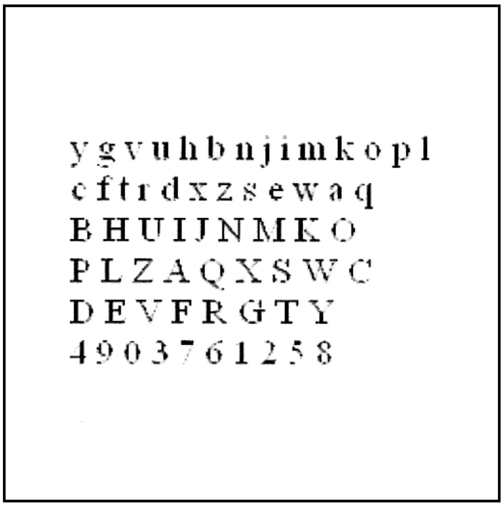
7 9 4 3 8 1 0 2 5 6
q w e a s d z x c r t
y f g h v b n u i o p l
k j m
P O I U J K L
M Y T R F G H
N B V C X Z D
S A E W Q

Text 4



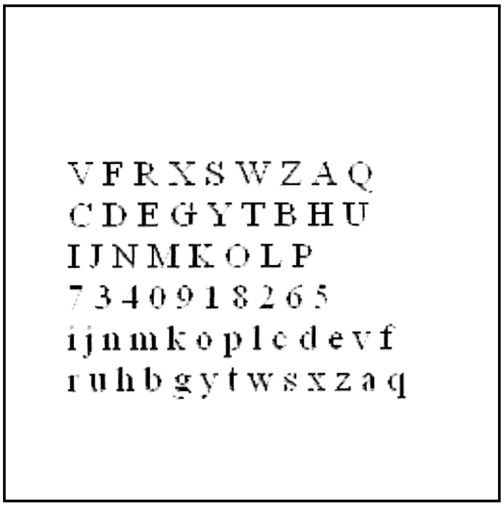
v b n m c x z g h j k l f d
s a y u i o p t r e w q
4 5 8 9 1 0 3 2 7 6
T Y U I O P R E W
Q G H J K L F D S
A B N M V C X Z

Text 5



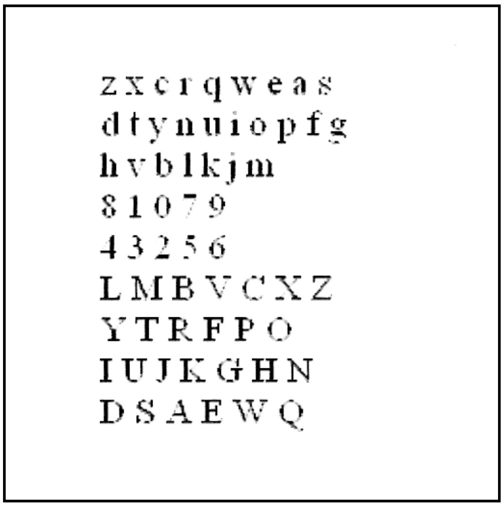
y g v u h b n j i m k o p l
c f t r d x z s e w a q
B H U I J N M K O
P L Z A Q X S W C
D E V F R G T Y
4 9 0 3 7 6 1 2 5 8

Text 6



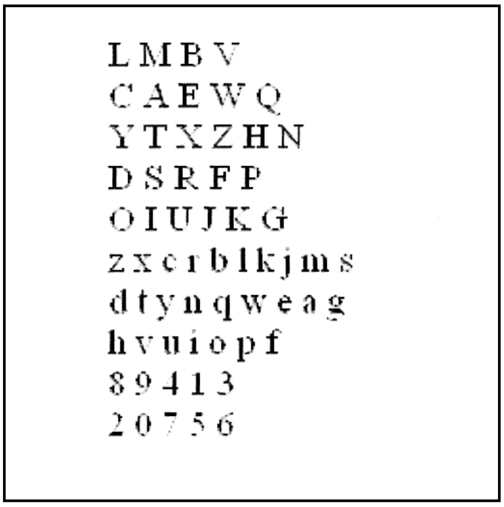
V F R X S W Z A Q
C D E G Y T B H U
I J N M K O L P
7 3 4 0 9 1 8 2 6 5
i j n m k o p l c d e v f
r u h b g y t w s x z a q

Text 7



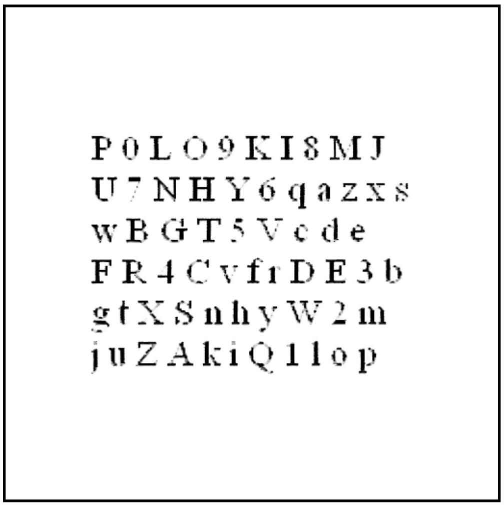
z x c r q w e a s
d t y n u i o p f g
h v b l k j m
8 1 0 7 9
4 3 2 5 6
L M B V C X Z
Y T R F P O
I U J K G H N
D S A E W Q

Text 8



L M B V
C A E W Q
Y T X Z H N
D S R F P
O I U J K G
z x c r b l k j m s
d t y n q w e a g
h v u i o p f
8 9 4 1 3
2 0 7 5 6

Text 9



P 0 L O 9 K I 8 M J
U 7 N H Y 6 q a z x s
w B G T 5 V c d e
F R 4 C v f r D E 3 b
g t X S n h y W 2 m
j u Z A k i Q 1 l o p

Text 10

List of Abbreviations

ARR	Average Recognition Rate
BMP	Bit-Map image file
DB	Database
DCT	Discrete Cosine Transform
DWT	Discrete wavelet Transform
E.D.	Euclidean Distance
HH	A signal or an image that has been high-pass filtered in both horizontal and vertical direction
HL	A signal or an image that has been vertically high-pass filtered and horizontally low pass filtered
HWT	Haar Wavelet Transform
IDCT	Inverse Discrete Cosine Transform
IHWT	Inverse Haar Wavelet Transform
LH	A signal or an image that has been horizontally high-pass and vertically low-pass filtered
LL	A signal or an image that has been high-pass filtered in both horizontal and vertical directions
OCR	Optical Character Recognition
WT	Wavelet Transform

List of Symbols

g	high pass filter
h	low pass filter
I_{xy}	Original image
N	Number of pixels
Y_{high}	Output of high pass filter
Y_{low}	Output of low pass filter
u	Translation Parameter
v	Feature Vector
x	Horizontal Coordinates
y	Vertical Coordinates
s	Scaling Parameter
$\varphi(t)$	Wavelet function