COMPARISON OF OCR ALGORITHMS USING FOURIER AND WAVELET BASED FEATURE EXTRACTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖNDER NAZIM ONAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

FEBRUARY 2011

Approval of the thesis:

COMPARISON OF OCR ALGORITHMS USING FOURIER AND WAVELET BASED FEATURE EXTRACTION

submitted by **ÖNDER NAZIM ONAK** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız ________________
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen ________________
Head of Department, **Scientific Computing**

Assist. Prof. Dr. Hakan Öktem ________________
Supervisor, **Department of Scientific Computing**

Examining Committee Members:

Prof. Dr. Gerhard Wilhelm Weber ________________
Institute of Applied Mathematics, METU

Assist. Prof. Dr. Hakan Öktem ________________
Institute of Applied Mathematics, METU

Assist. Prof. Dr. İlkay Ulusoy ________________
Department of Electrical and Electronics Engineering, METU

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÖNDER NAZIM ONAK

Signature :

ABSTRACT

COMPARISON OF OCR ALGORITHMS USING FOURIER AND WAVELET BASED FEATURE EXTRACTION

Onak, Önder Nazım
M.S., Department of Scientific Computing
Supervisor : Assist. Prof. Dr. Hakan Öktem

February 2011, 79 pages

A lot of research have been carried in the field of optical character recognition. Selection of a feature extraction scheme is probably the most important factor in achieving high recognition performance. Fourier and wavelet transforms are among the popular feature extraction techniques allowing rotation invariant recognition. The performance of a particular feature extraction technique depends on the used dataset and the classifier. Different feature types may need different types of classifiers. In this thesis Fourier and wavelet based features are compared in terms of classification accuracy. The influence of noise with different intensities is also analyzed. Character recognition system is implemented with Matlab. Isolated gray scale character image first transformed into one dimensional function. Then, set of features are extracted. The feature set are fed to a classifier. Two types of classifier were used, Nearest Neighbor and Linear Discriminant Function. The performance of each feature extraction and classification methods were tested on various rotated and scaled character images.

Keywords: Character recognition, Fourier transform, Wavelet transform, Ring projection

ÖZ

FOURIER VE DALGACIK TABANLI ÖZNİTELİK ÇIKARMA YÖNTEMLERİ KULLANARAK OPTİK KARAKTER TANIMA ALGORİTMALARININ KARŞILAŞTIRILMASI

Onak, Önder Nazım
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Assist. Prof. Dr. Hakan Öktem

Şubat 2011, 79 sayfa

Optik karakter tanıma alanında bircok araştırma sürdürülmektedir. Öznitelik seçimi yüksek tanıma performansı elde etmede muhtemelen en önemli etkendir. Fourier ve dalgacık analizleri popüler öznitelik çıkarma yöntemleri arasındadır. Bununla birlikte öznitelik çıkarma metodunun performansı kullanılan veri kümesi ve sınıflandırıcı tipine bağımlıdır. Bu tezde Fourier ve dalgacık analizine dayalı öznitelik çıkarma yöntemleri sınıflandırma doğrulukları temel alınarak karşılaştırılmıştır. Buna ek olarak çeşitli yoğunluktaki gürültünün etkiside gözlemlenmiştir. Gri tonlamalı karakter görüntüsü önce bir boyutlu fonksiyona dönüştürüldü ve öznitelikler çıkartılarak sınıflandırıcıya verildi. Sınıflandırıcı olarak En Yakın Komşu ve Doğrusal Ayırtedici Fonksiyonlar kullanılmıştır. Öznitelik çıkarma ve sınıflandırma yönteminin performansı çeşitli ölçekte ve döndürülmüş karakter görüntüleri kullanılarak test edilmiştir.

Anahtar Kelimeler: Karakter tanıma, Fourier dönüşümü, Dalgacık dönüşümü, Halka izdüşümü

To my nephew Doğa Kandemir

ACKNOWLEDGMENTS

I would like to thank all those people who have helped in the preparation of this study.

First of all, I am so grateful to my supervisor Assist. Prof. Dr. Hakan Öktem for introducing me wavelet analysis and his guidance through this study from initial to final level.

I would like to thank to my friend İsmail Uğur Tiryaki for his encouragement, comments and support.

I am grateful to my friend Ayşe Yıldız for her invaluable friendship and for her support.

Finally, I would like to special thanks to my mother Nuran Önol, my sister Nilüfer Onak Kandemir and her husband Özer Kandemir for their continuous support, patience and love.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Optical character recognition (OCR) has been very active research area since 1950's. It is one of the most successful application of automatic pattern recognition [54]. Its aim is to classify scanned, recorded or pictured images of machine printed or handwritten text, numerals, letters, symbols and generate a description in the desired format. Optical character recognition has many application area, digital libraries, packaging industries, personal digital assistants (PDAs). However, there are many problems that arise from the application areas. For example, in manufacturing line reading bottle caps as the bottles moving through the manufacturing line: The bottles rotate continuously as they move down in this line. Thus the character set printed on the caps, which is to be recognized by the machine vision system. There is no guarantee that the character set will be presented at a fixed orientation, precise location in front of the camera. At the same time all the printed characters may not be in same size. In order to overcome these drawbacks, OCR system must have capable of recognizing characters independent of their position, orientation and size.

General structure of OCR system is composed of *preprocessing, feature extraction, recognition* units. Selection of feature extraction method is probably the single most important factor in achieving high recognition performance [54]. Considering the fact that a very wide range of feature extraction methods exists in the literature, an important problem is deciding a suitable method for a particular application. Fourier and wavelet transforms are two popular feature extraction methods verified in many applications. However, which method performs better depends on the used dataset. In addition, OCR system performance also depends on the type of classifier used. Because, different feature types may need different types of classifier and vice versa.

This thesis is concentrated on translation, rotation and size invariant machine typed character recognition. Efficiency of Fourier and wavelet based feature extraction methods also evaluated in terms of classification accuracy. In addition, convenience of classifiers Nearest Neighbor and Linear Discriminant Functions were also tested.

This thesis is divided into seven chapter. In Chapter 2, history and general overview of OCR systems and review of some algorithms and techniques related to the preprocessing, feature extraction and classification methods are briefly explained.

Chapter 3, introduces the mathematical background of the study. We have used Fourier and wavelet transforms to extract and compose the character features.

Chapter 4, explains the details of proposed translation, rotation and size invariant character recognition system and its implementation.

Chapter 5, represents the experimental results obtained for each feature extraction (Fourier, wavelet) algorithms and classification (Nearest Neighbor, Linear Discriminant Function) methods.

Chapter 6, represents the character fonts used in our experiments and also recognition rate of each character obtained in each experiment.

Finally, concluding remarks are given in Chapter 7.

CHAPTER 2

OVERVIEW OF OPTICAL CHARACTER RECOGNITION

Documents are in the form of papers which the human can read and understand but it is not possible for the computer to understand these documents directly. Optical character recognition deals with the problem of classifying scanned images of machine printed or handwritten text, numerals, letters and symbols into a computer processable format such as ASCII without any human intervention. Useful reviews are found in [4, 41, 54]. Recognition is accomplished by searching a match between the features extracted from the given character image and the library of image models [11].

Character recognition systems can be classified based on upon two major criteria: the data acquisition process (online or offline) and the text type (machine printed or handwritten) [4]. However, online character recognition is sometimes confused with Optical Character Recognition. OCR refers to offline character recognition, where the system recognizes the fixed static shape of the character image, while online character recognition instead recognizes the dynamic motion during handwriting.

- Online Character Recognition: As the name infers online character recognition is real time recognition of characters. Online systems have better information for performing recognition since they have timing information and avoid the initial search step of locating the character as in the case of their offline counterpart. Online systems obtain the position of the pen as a function of time directly from the transducer that captures the writing as it is written. The most common of these devices is the electronic tablet or digitizer. Online recognition of characters is known as a challenging problem because of the complex character shapes and great variation of character symbols written in different modes [3, 4].

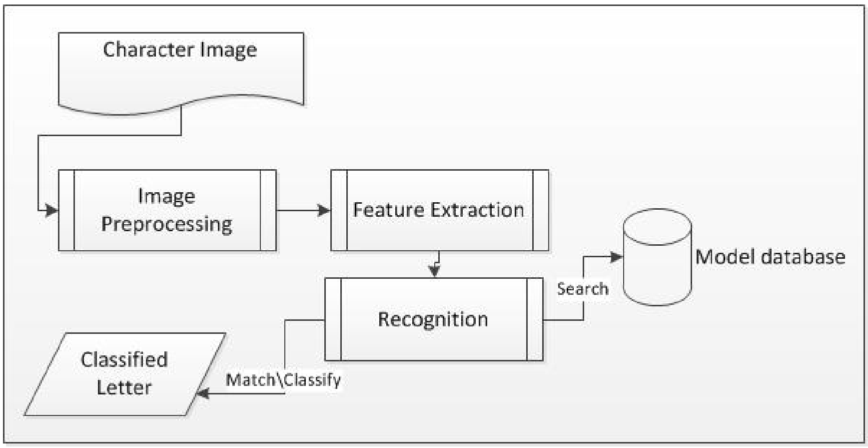


Figure 2.1: Typical optical character recognition system.

- Offline Character Recognition: Offline character recognition is known as Optical Character Recognition, because typewritten or handwritten character is scanned by optical scanner or camera and converted into form of a binary or gray scale image. Offline character recognition is a more challenging and difficult task as there is no control over the medium and instrument used. The artifacts of the complex interaction between the instrument medium and subsequent operations such as scanning and binarization present additional challenges to the algorithm for the offline character recognition. Therefore offline character recognition is considered as a more challenging task then its online counterpart [3, 4].

The major difference between Online and Offline Character Recognition is that Online Character Recognition has real time contextual information but offline data does not. This difference generates a significant divergence in processing architectures and methods.

No matter which text type (typewritten or handwritten) the problem belongs, OCR system can be roughly divided into image pre-processing (may include binarization, segmentation and conversion to other character representation), feature extraction and selection, classification stages. Figure 2.1 represents the basic OCR system. In this study we concentrated on machine printed optical character recognition problem. In typewritten character recognition, the images to be processed are in the forms of standard fonts like *Times New Roman, Arial, Courier*, etc..

2.1 Optical Character Recognition History

Dawn of OCR: In 1929 Gustav Tauschek obtained a patent on OCR in Germany, followed by Handel who obtained a US patent on OCR in USA in 1933. Tauschek's principle was template/mask matching which reflects the technology that time. His machine was a mechanical device that used photo detector and mechanical template matching. He was followed by Handel who obtained a US Patent on OCR in USA in 1933 (U.S. Patent 1,915,993).

Beginning of Research: In the beginning stage it was thought that it would be easy to develop an OCR. The 1950's and early half of the 1960's were periods when researchers imagined an ideal OCR, even through they were aware of the great difficulty of the problem. Actually, this is an instance of a common phenomenon which occurred in the research field of artificial intelligence [41]. The first modern character recognizers appeared in the middle of the 1940's with the development of the digital computers. The early work on the automatic recognition of characters has been concentrated either upon well printed text or upon small set of well distinguished hand written text. The first commercial system was installed at the Reader's Digest in 1955. The United States Postal Services has been using OCR machines to sort mail since 1965 based on technology devised primarily by the prolific inventor Jacob Rainbow. In 1965 it began planning an entire banking system, National Giro, using OCR technology, a process that revolutionized bill payment systems in the UK. The studies until 1980's suffered from the lack of advanced algorithms, powerful hardware and optical devices. With the explosion on the computer hardware and software technology, the previously developed methodologies found a very fertile environment for rapid growth in many application areas as well as OCR system development [3].

Current Status: The popularity of OCR is increasing each year with the advent of fast microprocessors, and better scanning technologies providing the vehicle for vastly improved recognition techniques.

2.2 Application of Character Recognition System

A common goal in the field of artificial intelligence is to mimic the function of human beings and automate tasks that normally require manual labor or at least supervision. Optical character recognition is a classic example of such a problem. It has been the subject of a large body of research because there are great number of commercial applications for this technology. Some of the most significant applications include

- *Address Reading:* Address readers are used by a postal mail services which sort and locate the destination block and postal code this address block then mails are grouped depending on destinations for delivery purposes.

- *Form Reading:* A form reading system discriminate between pre-printed form instructions and filled in data. The system is first trained with a blank form. The system, registers those areas on the form where the data should be printed. During the form recognition phase, the system uses the spatial information obtained from training to scan the region that should be filled with data.

- *Check Reader:* A check reader captures check image and recognize amounts and accounts information on the checks and use the information in both fields to cross check the recognition result. An operator intervention might require to correct misclassified characters by cross-validating the recognition results with the check.

- *Bill Processing:* Bill processing system is used to read payment slips, utility bills and inventory documents .The system focuses on certain region on a document where the expected information are located, e.g. account number and payment value.

- *Page Readers:* Page reader applications have used for various purposes. One of the application is converting non editable written document, which might be stored in a image format, into ASCII file. Converted file then can be processed depending on application area. Another innovative application is conversion of scanned pages into spoken words.

- *License Plate Recognition:* License plate recognition applies image processing and character recognition technology to identify vehicles by automatically reading their license plates. Typical applications of license plate recognition include private parking lot management, traffic monitoring, automatic traffic ticket issuing, automatic toll payment, surveillance, and security enforcement.

2.3 Steps of Optical Character Recognition System

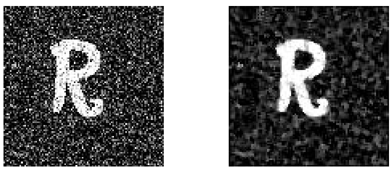
This section focus on the machine printed character recognition methodologies. The literature review of OCR point out that the task of OCR system can be grouped in the stage of preprocessing, feature extraction and selection and recognition. In various applications some of the stages are merged or excluded. In this section, the methodologies to develop the stages of the OCR will be explained.

2.3.1 Preprocessing

Pre-processing is the name given to a family of procedures,which will contribute in defining a compact representation of pattern, along the road to final classification can be made simple and more accurate. Depending on the application and techniques number of steps in preprocessing stage can be change. Nevertheless, the following techniques are common most of the character recognition applications.

2.3.1.1 Noise Filtering

The objective of noise filtering is to remove any unwanted bit-patterns, and suppress spurious points usually introduced by writing surface or poor sampling rate of the image acquisition device. The basic idea is convolving the image with a pre-defined mask in order to assign



(a) Noisy. (b) Filtered.

Figure 2.2: Illustration of noisy and noise filter applied character images.

a value to a pixel as a function of the gray values of its neighboring pixels. Let $x(i, j)$ be the noisy image under consideration, then the basic linear filtering process defined as follows:

$$x_{filtered}(i, j) = \sum_k \sum_l a_{kl} x(i - k, j - l), \tag{2.1}$$

where a_{kl} is the weight of the gray levels of pixels of the mask at location (k, l). Sample noisy and fitered images are given in Figure 2.2.

2.3.1.2 Segmentation

Segmentation refers to the decomposition of image into its components in order to make the image easier to analyze. It is important stage in image analysis which directly affects recognition rates. Character segmentation is the process in which from the word segmentation we extract only characters. Segmentation can be divided into two type *external* and *internal*. External segmentation decomposes the image into its logical units. For example, in document analysis, page layout is decomposed into paragraphs, sentences and words. Internal segmentation is an operation that seeks to decompose and image into individual symbols, characters, shapes, etc.. Some of segmentation techniques are listed below and more information can be found in [28].

- Amplitude thresholding or window slicing.
- Component labeling.
- Boundary based approaches.
- Clustering and region based approaches.
- Template matching.
- Texture segmentation.

2.3.1.3 Binarization

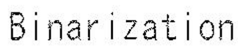
Image binarization refers to the conversion of a gray-scale image into a binary image. Each pixel in an image is converted into one bit and the value '1' or '0' assigned depending upon the value of the pixel and threshold value. Binarization process is categorized into two group global and local.

- Global binarization, picks one threshold value for the entire image which is often based on an estimation of the background level from the intensity histogram of the image.

Figure 2.3 represents the gray scale image and binary image obtained using global binarization method.

- Local binarization, uses different threshold values for each pixel according to the local area information.

(a) Gray scale image.

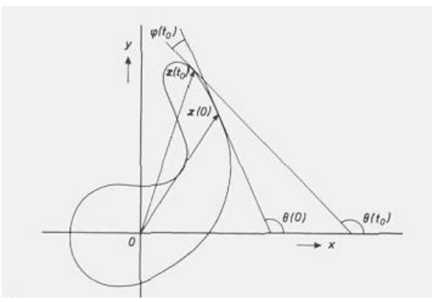


(b) Binary image.

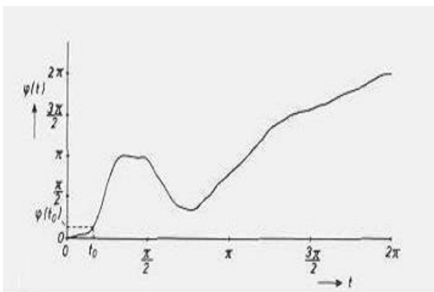
Figure 2.3: Binarization.

2.3.1.4 Conversion to Other Representation

In some recognition application, character patterns are transformed to other representation models. Zahn [57] proposed a method which represent the contour of shape or character image as a function cumulative angular function $\phi(t)$, that represents the net angular bend between starting point and point t of contour (Figure 2.4). The $\phi(t)$ representation measures the way in which the shape in question differs from the circular shape. Another method is proposed by Tang in [49] which transform two dimensional binary pattern into one dimensional function using rings. We will explain details of Tang's projection method in the 4th chapter.



(a) Contour.



(b) Cumulative angular function ϕ.

Figure 2.4: Illustration of the shape contour and cumulative angular function.

2.3.2 Feature Extraction and Selection

A feature is a distinctive or characteristic measurements, transform, structural component extracted from a segment of a pattern [17] which is used to describe patterns by intending to minimize loss of any important information. It is important to make distinction between feature extraction and selection. However these terms are used interchangeably in literature [29]. Feature extraction is a process of determining feature vector of given pattern and it is one of the most important step in pattern recognition applications. Because, it is required to convert pattern into feature vector in order to make the recognition problem solvable.

Definition 2.3.1 *Feature extraction: extraction from the raw data the information which is more relevant for classification purposes, in the sense of minimizing the within-class pattern variability while enhancing the between-class variability.*

Feature selection refers to a process of selecting best subset from future set.

Definition 2.3.2 *Given a set of features $\vec{x} \in \mathbb{R}^d$, select a subset of size $\vec{\vec{x}} \in \mathbb{R}^m$ where $m \leq d$, that leads to the smallest classification error.*

Selecting features from set of measurements decreases the size of feature vector which improves classification performance. The reasons for the necessity to decreasing feature dimension can be summarized as follows. Reducing computational complexity, data storage requirement, training and utilization times. In addition, some of the extracted features might be irrelevant, redundant or sensitive to distortion. Thus, it is required to eliminate those from whole set.

There are many feature extraction techniques described in literature. These are include, Fourier descriptors [10, 13, 57], Wavelet descriptors [13, 17, 50], Zernike moments [33], Independent component analysis [42], Zoning [21]. However, efficiency of each feature extraction method varies depending on the specific recognition problem and available data. In addition, it is not easy to compare performance of feature extraction methods reported in the literature, because they are based on different data sets [54]. These reasons make the selection of best feature extraction method challenging task for given specific application.

- *Fourier Transform:* The common practice is to use the magnitude spectrum of measurement vector as the feature set. One of the most attractive property of the Fourier transform is its position shift invariance if we only consider magnitude spectrum and ignore the phase.

- *Wavelet Transform:* The wavelet transform is a series expansion technique that provide us to represent the signal or image in different resolution levels which is called multiresolution representation. At different resolution, the details of images characterize different physical structures [36].

- *Template Matching:* In this method the character image itself is used as a feature vector. Similarity between the input and each template image is computed by comparing pixels of the binary image [54]. Jaccard and Yula distances are two of the similarity measurement methods. Let n_{ij} be the number of pixel positions where the template pixel x is i and the image pixel y is j, with $i, j \in 0, 1$

$$n_{ij} = \sum_{m=1}^{n} \delta_m(i, j), \tag{2.2}$$

where

$$\delta_m(i, j) = \begin{cases} 1, & if \quad (x_m = i) \wedge (y_m = j), \\ 0, & otherwise, \end{cases} \tag{2.3}$$

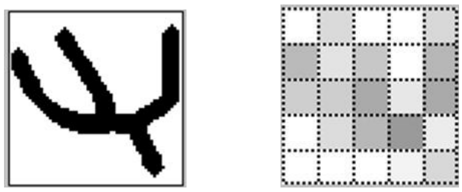
$$d_J = \frac{n_{11}}{n_{11} + n_{10} + n_{01}}, \tag{2.4}$$

$$d_Y = \frac{n_{11}n_{00} - n_{10}n_{01}}{n_{11}n_{00} + n_{10}n_{01}}, \tag{2.5}$$

- *Zoning:* The character image is divided into *mxn* zones then percentage of black pixels analyzed as features Figure 2.5. Additional features can be used to improve performance such as contour directions and strokes [21, 54].

- *Crossing and Distances:* Features are measured from the crossings count the number of transitions from background to foreground pixels along vertical and horizontal directions through the character image and distances calculate the distances of the first image pixel detected from the upper and lower boundaries, along vertical and from the left and right boundaries along horizontal lines (Figure 2.6) [22].

- *Projection Histogram:* Projection histograms count the number of pixels in each column and row of a character image Figure 2.7. Let $y(i)$ represent the horizontal projec-



(a) Image and divided zones.

(b) Density of zones.Darker squares indicate higher density of zone pixels.

Figure 2.5: Zoning method.

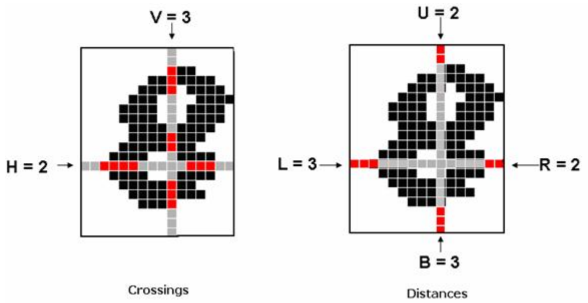


Figure 2.6: Crossing and distances.

tion, dissimilarity between two character image is given as follows [54]:

$$D = \sum_{i=1}^{n} |Y_1(i) - Y_2(i)| , \tag{2.6}$$

where

$$Y(i) = \sum_{j=1}^{i} y(j), \tag{2.7}$$

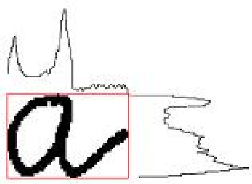


Figure 2.7: Vertical and horizontal histogram.

- *Zernike Moments:* Zernike moments are projection of the input image onto space spanned by the V functions which form a complete orthonormal set over the unit disk of $x^2 + y^2 \leq 1$ in polar coordinates. The fundamental feature of Zernike moments is their rotational invariance and the magnitudes of Zernike moments can be used as rotationally invariant image feature [33, 54].

- *Moment Invariants:* An image can be represented as a point in N dimensional vector space by using N moments [20, 54]. This converts the pattern recognition problem into a standard decision theory problem. Let $\mathcal{R}$ be the represent the region of pattern to be recognized then $(p + q)$-th order moment of region is defined as follows:

$$\int\int_{\mathcal{R}} x^p y^q dxdy \qquad (p, q = 1, 2, 3, ...), \tag{2.8}$$

2.3.3 Classification Methods

After feature extraction and selection procedure completed, a classifier can be designed using several number of available approaches. A classifier is a function that takes the various of features in an example and predicts the class that example belongs to. It is not easy problem to choose classifier, because there is no single optimal approach for classification. There are many admissible classifier design approaches available, each capable of discriminating patterns in certain portion of the feature space. In supervised learning, no matter which classification approach is used, classification process operate in two mode:training and classification [29]. In training mode, classifier is trained based on measured features of each known class patterns. The learned classifier is basically a model of the relationship between the features and the class label in the training set. In classification mode, the trained classifier assigns the input pattern to one of the pattern classes considering measured features. Consequently the performance of the classifier depends on both available training samples as well as the specific values of the samples. The idea is that, if the classifier truly captured the relationship between features and classes, it ought to be able to predict the classes of examples [44]. The are four main recognition approaches:

- Template Matching.
- Statistical Approach.
- Syntactic or Structural Approach.
- Neural Networks.

The above techniques are not necessarily independent and same recognition method may exist with different interpretations.

2.3.3.1 Template Matching

It is one of the earliest and simplest approaches to pattern recognition, which is used to determine the analogy between two entities of the same type. In template matching, the pattern to be recognized is matched against the stored template while taking into account all allowable changes. Template matching is computationally demanding and it would be fail if the patterns are distorted due to the imaging process, viewpoint change, or large interclass variations among the patterns [29].

2.3.3.2 Statistical Approach

In statistical pattern recognition,given a set of measurements obtained through observation and represented as a d dimensional pattern vector x, pattern is assigned to one of C possible classes ($\omega_i \quad i = 1, ..., C$). The features are assumed to have a probability density function conditioned on the pattern class and feature vector x is viewed as an observation drawn randomly from the class conditional probability function $p(x|w_i)$. A decision rule partitions the measurement space into C regions ($\Omega_i \quad i = 1, ..., C$). If an observation vector is in Ω_i then it is assumed to belong to class ω_i.

Depending on the available information about the class conditional densities various strategies are utilized in order to design classifier. If class conditional densities are known Bayes decision rule can be used. On the other hand, it is not common situation in practice. In such a case two different strategy can be followed: parametric and nonparametric. Parametric approach assumes that the form of the class conditional densities are known, but some parameters of densities need to be estimated using training data. However, if the form of the class conditional densities is not known it is required to estimate the density function or directly construct the decision boundary. Detailed information about statistical recognition methods can be found in [19, 56].

2.3.3.3 Syntactic or Structural Approach

In many recognition problem it is appropriate to view pattern as being composed of simpler sub patterns. The elementary sub patterns are called primitives and any given complex pattern can be represented in terms of relationship between these primitives. Syntactic or Structural pattern recognition, provides a description of how the given pattern is constructed from the primitives. This paradigm has been used in situations where the patterns have a definite structure which can be captured in terms of a set of rules [29].

2.3.3.4 Neural Networks

A neural network is a graph, with patterns represented in terms of numerical values attached to the nodes, which can be called neurons, of the graph and transformations between patterns achieved via message-passing algorithms. The nodes in the graph are divided as input nodes or output nodes, and the graph as a whole can be viewed as a representation of a parallel multivariate function linking inputs to outputs. All links have a weights which parameterizing the input and output function and they are adjusted by learning algorithm. Since it has a parallel architecture computations can be made in higher rate compared to classical techniques. In addition, neural networks can adapt to changes in data and learn the characteristic of the input.

Neural network architecture involves treating the network as a statistical processor, characterized by making particular probabilistic assumptions about data. Patterns appearing on the input nodes or the output nodes of a network are viewed as samples from probability densities, and a network is viewed as a probabilistic model that assigns probabilities to patterns. The problem of learning the weights of a network is thereby reduced to a problem in statistics that of finding weight values that look probable in the light of observed data [30].

2.4 Factors Affecting OCR Accuracy

There are a number of key factors to consider when observing a printed resource and assessing whether it will produce the text resource accuracy desired through OCR technologies. Some of the main ones are listed below:

2.4.1 Scanning methods

If the image can be represented as gray scale or better, then this is more likely to improve the OCR accuracy than almost any other scanning mechanism. Gray scale is clearly readable thus, the lower the standard of scanned image then the worse the OCR accuracy is likely to be. There are other factors related to the nature of the original that will affect the standard of the scanned image and these should be accounted for as well.

2.4.2 Nature of original paper

The original paper on which text appears is critical to the scanned image. It is very difficult to read text which appears against a very dark background, or is printed over words or graphics. Programming a system to interpret only the relevant data and disregard the rest is a difficult task. If the OCR engine cannot discriminate between the character and the paper background noise then it will be more likely to misrepresent the character. Gray scale images as opposed to black and white give the OCR software a better chance of discriminating between text and noise and thus improve the accuracy.

2.4.3 Nature of printing

The nature of the printed text in the original may make a significant difference to OCR accuracy. Obviously if the text is printed poorly or if it was typed and characters are broken, faded or have indistinct edges then this will affect the ability of an OCR engine to recognize patterns and differentiate between similar shaped characters. So the clarity of the printing is a factor to consider. For example, while examining a string of characters combining letters and numbers, there is very little visible difference between a capital letter ”O” and the numeral ”0”. Humans can re-read the sentence or entire paragraph to determine the accurate meaning. This procedure, however, is much more difficult for a machine. Some fonts may also have improved print clarity over others and also by the use of larger point sizes.

2.4.4 Formatting complexities

Variations in font size and type face may result in misunderstanding the characters. Broken character and touching character stemming from excess ink or paper degradations may not be recognized.

CHAPTER 3

PRELIMINARIES

3.1 Mathematical Background

3.1.1 The Fourier Transform

The Fourier transform was developed by Baron Jean-Baptiste-Joseph Fourier (1768-1830), a French mathematician and physicist, and he applied it in order to explain many instances of heat conduction. Fourier transform provides an alternative way of representing data by decomposing a function in space or time into its various frequency components. The basic building blocks are sine and cosine functions which vibrate at a frequency of n times per 2π interval. Each frequency component have its own amplitude and phase information where amplitude refers the magnitude of the sinusoidal and phase specifies the starting point in the cycle. Fourier transform can be applied to both continuous and discrete functions.

3.1.1.1 The Continuous Fourier Transform

Theorem 3.1.1 *If $f : \mathbb{R} \to \mathbb{R}$ is a continuously differentiable function with $\int_{-\infty}^{\infty} f(t)dt < \infty$, then the Fourier transform of f is given by*

$$\widehat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt, \tag{3.1}$$

and inverse Fourier transform can be obtained by

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \widehat{f}(\omega)e^{i\omega t}dt, \tag{3.2}$$

where ω is angular velocity.

3.1.1.2 The Discrete Fourier Transform

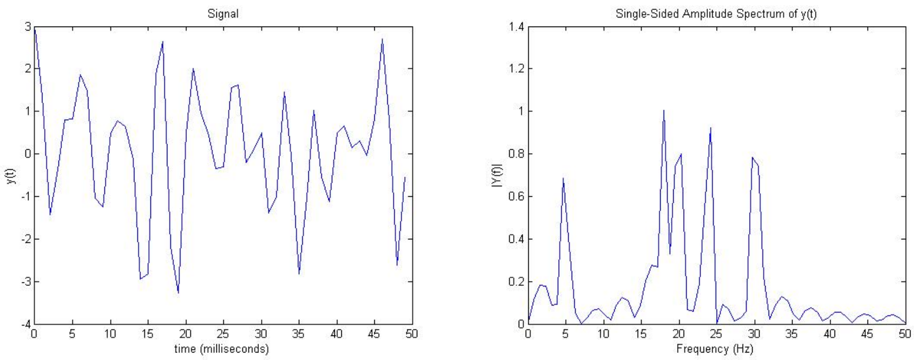
The discrete Fourier transform is used to find the frequency spectrum of sampled functions (Figure 3.1).

Definition 3.1.2 *Let $\{y_n\}_0^{N-1}$ be a sequence of N complex numbers the discrete Fourier transform (DFT) of y is the sequence $\widehat{y_n}$, where*

$$\widehat{y_k} = \sum_{n=0}^{N-1} y_n e^{\frac{-2\pi ikn}{N}} \qquad (k = 0, ..., N-1)\,, \tag{3.3}$$

The inverse discrete Fourier transform is given by

$$y_n = \frac{1}{N}\sum_{n=0}^{N-1} \widehat{y_k} e^{\frac{2\pi ikn}{N}} \qquad (n = 0, ..., N-1)\,, \tag{3.4}$$



(a) A sample function. (b) Amplitude spectrum.

Figure 3.1: A sample function and its single sided amplitude spectrum.

3.1.1.3 Basic Properties

In this section we will give some basic properties of the Fourier transform. We will use notation $\mathcal{F}\left[f(t)\right](\omega) = \widehat{f}(\omega)$ and $\mathcal{F}^{-1}\left[\widehat{f}(\omega)\right](t) = f(t)$ to represent Fourier transform and its inverse:

$$\mathcal{F}[af(t) + bg(t)](\omega) = a\mathcal{F}[f(t)](\omega) + b\mathcal{F}[g(t)](\omega), \tag{3.5}$$

$$\mathcal{F}^{-1}\left[a\widehat{f}(\omega) + b\widehat{g}(\omega)\right](t) = a\mathcal{F}^{-1}\left[\widehat{f}(\omega)\right](t) + b\mathcal{F}^{-1}\left[\widehat{g}(\omega)\right](t), \tag{3.6}$$

$$\mathcal{F}\left[t^n f(t)\right](\omega) = i^n \frac{d^n}{d\omega^n}\mathcal{F}\left[f(t)\right](\omega), \tag{3.7}$$

$$\mathcal{F}^{-1}\left[\omega^n \widehat{f}(\omega)\right](t) = (-i)^n \frac{d^n}{dt^n}\mathcal{F}^{-1}\left[\widehat{f}(\omega)\right](t), \tag{3.8}$$

$$\mathcal{F}\left[f(t-a)\right](\omega) = e^{-i\omega a}\mathcal{F}\left[f(t)\right](\omega), \tag{3.9}$$

$$\mathcal{F}\left[f(bt)\right](\omega) = \frac{1}{b}\mathcal{F}\left[f(t)\right](\frac{\omega}{b}), \tag{3.10}$$

$$\mathcal{F}[f(t) * g(t)](\omega) = \mathcal{F}[f(t)](\omega).\mathcal{F}[g(t)](\omega), \tag{3.11}$$

3.1.2 The Wavelet Transform

Wavelets are wave-like osilating functions that satisfy certain mathematical requirements. They have been introduced by A. Grossman and J. Morlet as function $\psi(x)$ whose translation and dilation $\left(\sqrt{s}\psi(sx-t)\right)_{(s,t)\in\mathbb{R}^+\times\mathbb{R}}$ can be used for expansion of $L^2(\mathbb{R})$ functions. J. Stromberg and Y. Meyer showed independently that there exist wavelets $\psi(x)$ such that whose translation and dilation $\left(\sqrt{2^j}\psi\left(2^j x-k\right)\right)_{(j,k)\in\mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$ [36, 37]. The function $\psi(x)$ has a companion, the scaling function $\phi(x)$. These two functions can be used to decompose or reconstruct the functions in $L^2(\mathbb{R})$. Scaling and wavelet functions satisfy the following relations:

$$\phi(x) = \sum_{k\in\mathbb{Z}} h_k\phi(2x-k), \tag{3.12}$$

$$\psi(x) = \sum_{k\in\mathbb{Z}} g_k\phi(2x-k), \tag{3.13}$$

where h_k and g_k are called lowpass and highpass filter coefficients respectively.

$$h_k = 2\left\langle\phi(x), \phi(2x-k)\right\rangle, \tag{3.14}$$

$$g_k = (-1)^k h^*_{1-k}, \tag{3.15}$$

3.1.2.1 Multiresolution Analysis

A multiresolution analysis of $L^2(\mathbb{R})$ is defined as a sequence of closed subspaces $V_j \subset L^2(\mathbb{R}), (j\in\mathbb{Z})$.

Definition 3.1.3 *Let $V_j, (j = \ldots, -2, -1, 0, 1, 2, \ldots)$ be a sequence of subspaces of functions in $L^2(\mathbb{R})$. The collection of subspaces $\left\{V_j, j\in\mathbb{Z}\right\}$ is called a multiresolution analysis with scaling function ϕ if the following conditions hold:*

- $V_j \subset V_{j+1}$.
- $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$ *and* $\bigcup_{j\in\mathbb{Z}} V_j$ *is dense in* $L^2(\mathbb{R})$.
- *The function* $f(x)$ *belongs to* V_j *if and only if the function* $f\left(2^{-j}x\right)$ *belongs to* V_0.
- *The function* ϕ *is belongs to* V_0 *and the set* $\{\phi(x-k), k \in \mathbb{Z}\}$ *is orthonormal basis for* V_0.

The V_j's are called approximation spaces and different choices of ϕ may yield different multiresolution analysis.

Theorem 3.1.4 *Suppose* $\left\{V_j, j \in \mathbb{Z}\right\}$ *is a multiresolution analysis with scaling function* ϕ. *Then for any* $j \in \mathbb{Z}$ *the set of functions*

$$\phi_{jk}(x) = 2^{j/2}\phi(2^j x - k), \qquad (k \in \mathbb{Z}), \tag{3.16}$$

is an orthonormal basis for V_j.

Theorem 3.1.5 *Suppose* $\left\{V_j, j \in \mathbb{Z}\right\}$ *is a multiresolution analysis with scaling function* ϕ. *Then the following relation holds:*

$$\phi(x) = \sum_{k\in\mathbb{Z}} h_k \phi(2x-k) \qquad \textit{where} \qquad h_k = 2\int_{-\infty}^{\infty} \phi(x)\overline{\phi(2x-k)}, \tag{3.17}$$

Moreover, we also have

$$\phi(2^{j-1}x - l) = \sum_{k\in\mathbb{Z}} h_{k-2l}\phi(2^j x - k), \tag{3.18}$$

or, equivalently

$$\phi_{jk}(x) = 2^{-1/2}\sum_{k\in\mathbb{Z}} h_{k-2l}\phi_{jk}, \tag{3.19}$$

where $\phi_{jk}(x) = 2^{j/2}\phi(2^j x - k)$.

Theorem 3.1.6 *Suppose* $\left\{V_j, j \in \mathbb{Z}\right\}$ *is a multiresolution analysis with scaling function* ϕ. *Then the following equalities hold:*

$$\sum_{k\in\mathbb{Z}} h_{k-2l}\overline{h_k} = 2\delta_{l0}, \tag{3.20}$$

$$\sum_{k\in\mathbb{Z}} |h_k|^2 = 2, \tag{3.21}$$

$$\sum_{k\in\mathbb{Z}} h_k = 2, \tag{3.22}$$

$$\sum_{k\in\mathbb{Z}} h_{2k} = 1, \tag{3.23}$$

$$\sum_{k\in\mathbb{Z}} h_{2k+1} = 1, \tag{3.24}$$

Theorem 3.1.7 *Suppose $\left\{V_j, j \in \mathbb{Z}\right\}$ is a multiresolution analysis with scaling function*

$$\phi(x) = \sum_{k\in\mathbb{Z}} h_k \phi(2x - k), \tag{3.25}$$

Let $\left\{W_j\right\}$ be span of $\left\{\psi\left(2^j x - k\right), k \in \mathbb{Z}\right\}$, where

$$\psi(x) = \sum_{k\in\mathbb{Z}} (-1)^k h^*_{1-k} \phi(2x - k), \tag{3.26}$$

Then $W_j \subset V_{j+1}$ is the orthogonal complement of V_j in V_{j+1}. Furthermore $\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k), k \in \mathbb{Z}$ is an orthogonal basis for the W_j.

By successive orthogonal decomposition,

$$V_j = W_{j-1} \oplus W_{j-2} \oplus \ldots \oplus W_0 \oplus V_0, \tag{3.27}$$

$$= W_{j-1} \oplus W_{j-2} \oplus \ldots \oplus W_0 \oplus W_{-1} \oplus \ldots. \tag{3.28}$$

Equivalently, each $f \in L^2(\mathbb{R})$ can be uniquely expressed as a sum $\sum_{k=-\infty}^{+\infty} w_k, (w_k \in W_k)$ and where the w_k's are mutually orthogonal.

3.1.2.2 Decomposition and Reconstruction

Suppose that we have function f which is already in one of the approximation space, such as V_j:

$$f = \sum_{k\in\mathbb{Z}} \left\langle f, \phi_{jk}\right\rangle \phi_{jk}, \tag{3.29}$$

The one level decomposition of function f has the form

$$f = \underbrace{\sum_{k\in\mathbb{Z}} \left\langle f, \phi_{j-1,k}\right\rangle \phi_{j-1,k}}_{f_{i-1}} + \underbrace{\sum_{k\in\mathbb{Z}} \left\langle f, \psi_{j-1,k}\right\rangle \psi_{j-1,k}}_{w_{i-1}}, \tag{3.30}$$

where f_{i-1} , w_{i-1} are called approximation and detail of function f respectively. Reconstruction of function f is defined as:

$$\left\langle f, \phi_{jk}\right\rangle = 2^{-1/2} \sum_{l\in\mathbb{Z}} h_{k-2l} \left\langle f, \phi_{j-1,l}\right\rangle + 2^{-1/2} \sum_{l\in\mathbb{Z}} g_{k-2l} \left\langle f, \psi_{j-1,l}\right\rangle, \tag{3.31}$$

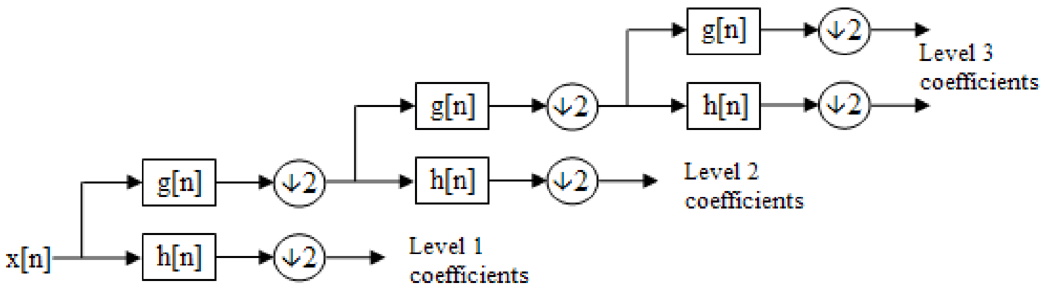
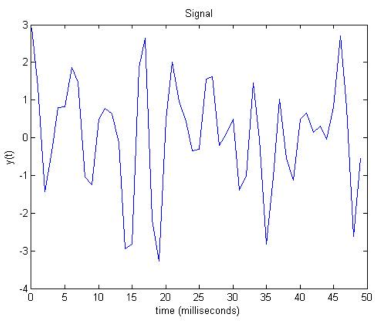
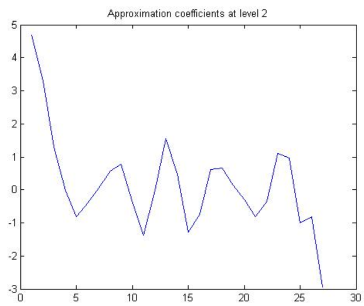


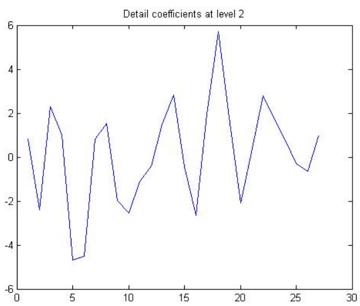
Figure 3.2: Wavelet decomposition.



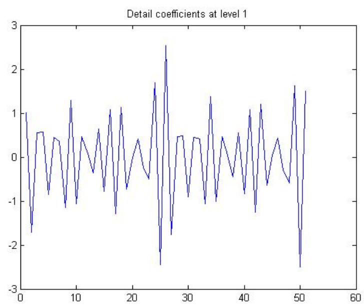
(a) A sample function.



(b) Approximation coefficients at level 2.



(c) Detail coefficients at level 2.



(d) Detail coefficients at level 1.

Figure 3.3: Decomposition of a sample function using Daubechies 2 wavelet.

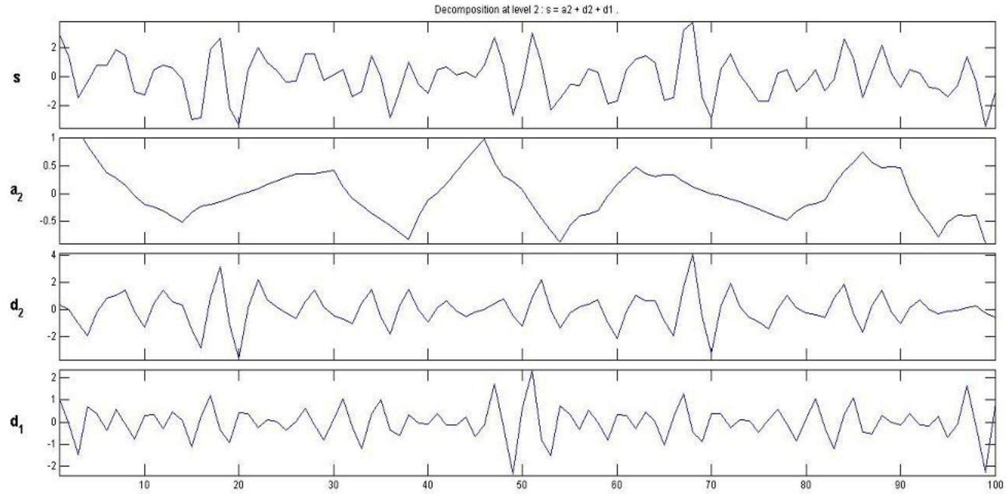


Figure 3.4: A sample function and its approximation, detail functions for 2 level Daubechies 2 wavelet decomposition.

3.1.2.3 Wavelet families

There are different types of wavelet families whose qualities vary according to several criteria. The main criteria are:

- The support of ψ and $\mathcal{F}\left[\Psi(t)\right](\omega)$: the speed of convergence to 0 of these functions when the time t or the frequency ω goes to infinity, which quantifies both time and frequency localizations.
- The symmetry, which is useful in avoiding dephasing in image processing.
- The number of vanishing moments for ψ or for ϕ if it exists, which is useful for compression purposes.
- The regularity, which is useful for getting nice features, like smoothness of the reconstructed signal or image, and for the estimated function in nonlinear regression analysis.

These are associated with two properties that allow fast algorithm and space-saving coding:

- The existence of a scaling function ϕ.
- The orthogonality or the biorthogonality of the resulting analysis.

There are several wavelet families available that have been proven to be useful in many application areas including, pattern recognition, image processing, signal processing, etc. The following section gives an brief information about the wavelet families that we used in our study except Haar wavelet.

- *Haar:* It was proposed in 1909 by Alfréd Haar (Figure 3.5). Haar used these functions to give an example of a countable orthonormal system for the space of square-integrable functions on the real line. The Haar wavelet is the simplest possible wavelet. The technical disadvantage of the Haar wavelet is that it is not continuous, and therefore not differentiable. This property can, however, be an advantage for the analysis of signals with sudden transitions:

$$\phi(x) = \begin{cases} 1, & 0 \leq x < 1, \\ 0, & elsewhere, \end{cases} \tag{3.32}$$

$$\psi(x) = \phi(2x) - \phi(2x - 1), \tag{3.33}$$

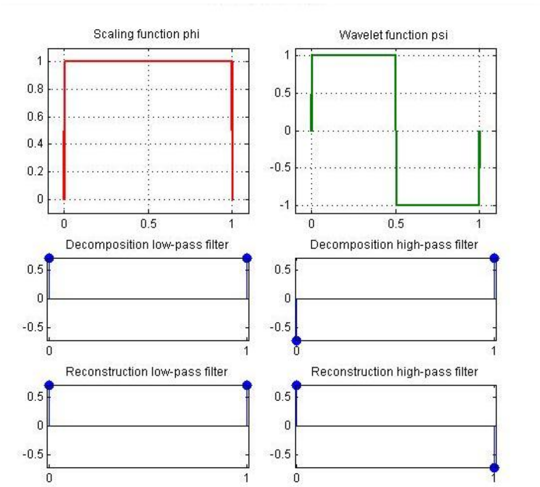


Figure 3.5: Haar wavelet scaling, wavelet functions and decomposition, reconstruction filter coefficients.

- *Daubechies wavelets:* Invented by Ingrid Daubechies (Figure 3.6). They are orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. With each wavelet type of this class, there is a scaling function (also called father wavelet) which generates an orthogonal multiresolution analysis.

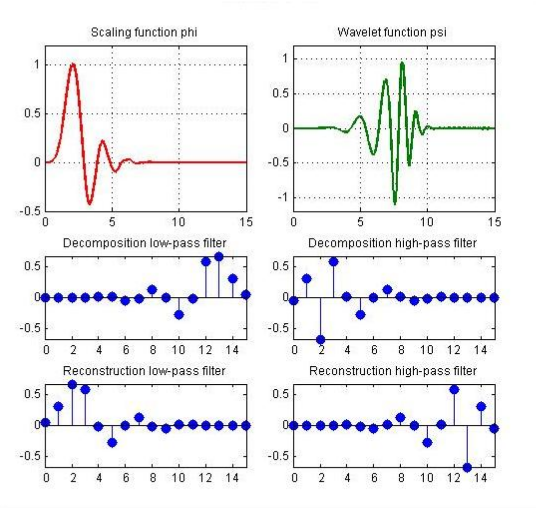


Figure 3.6: Daubechies 8 scaling, wavelet functions and decomposition, reconstruction filter coefficients.

- *Symlets:* The symlets are nearly symmetrical wavelets proposed by Daubechies as modifications to the Daubechies wavelet family (Figure 3.7).

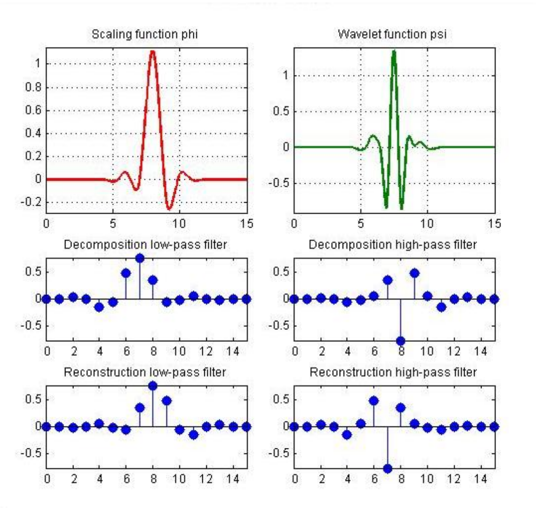


Figure 3.7: Symlet 8 scaling, wavelet functions and decomposition, reconstruction filter coefficients.

- *Coiflets:* Coiflets are discrete wavelets designed by Ingrid Daubechies (Figure 3.8), at the request of Ronald Coifman, to have scaling functions with vanishing moments. The wavelet is near symmetric, their wavelet functions have $N/3$ vanishing moments and scaling functions $N/3 - 1$.

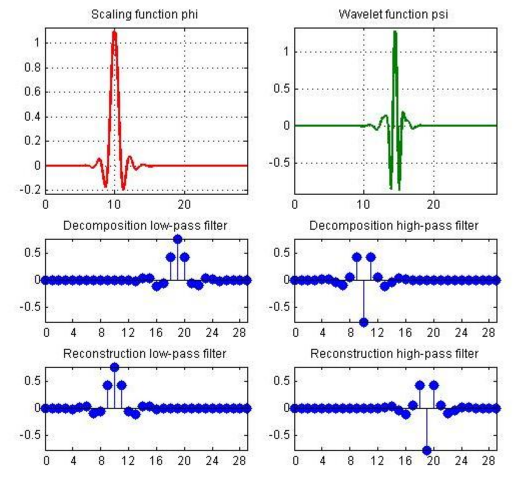


Figure 3.8: Coiflet 5 scaling, wavelet functions and decomposition, reconstruction filter coefficients.

CHAPTER 4

IMPLEMENTATION

This chapter describes a translation, scale and rotation invariant machine printed character recognition scheme. The scheme first apply noise filtering procedure to the given gray scale character image, then filtered image is converted into binary format. The binary image is transformed into another form using rings in order to obtain one dimensional translation, scale and rotation invariant representation. Then the features are extracted from projected character representation.

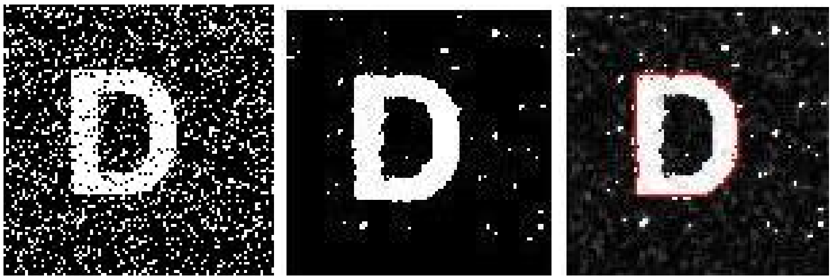
In this chapter, a detailed implementation of the character recognition model is described.

4.1 Recovering Occluded Character Image

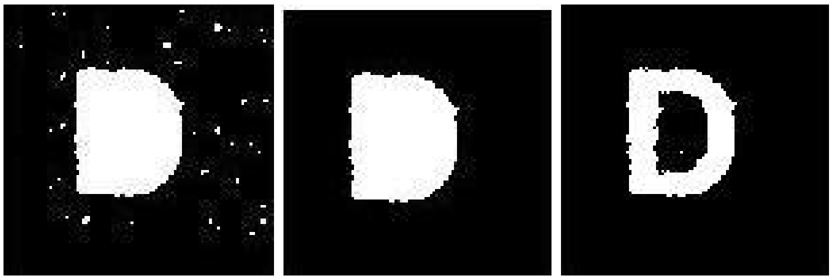
The occlusion problem means that a part of one object is obscured by another object [1]. In our study foreground occlusion were considered. Because we have seen that after noise filtering and binarization steps background has contained additional foreground colors if the noise intensity was high (Figure 4.1). Although the images we have used are graysclae format, the character recognition algorithm deals with black and white images. Everything white is threat as if belonging to the character is called foreground color. The rest is black and called background color.

If foreground occlusion occurs then there is occlusion because the foreground color is obscuring some of the background color. So there is additional foreground color attached which now belongs to the character [9]. Consequently, the character recognition system can take these additional foreground colors as a part of the character which cause to fail the algorithm.

Detecting and reconstructing of this type of occlusion is based on searching for character region entirely contained in the background, so that character region and background can be separated. Extracted background can be reconstructed changing white regions with black region. Figure 4.1 represents the occluded background and steps of recovering occluded character image.



(a) Noisy character image. (b) Occluded background after noise filtering and binarization. (c) Detected character boundary.



(d) Extracted background. (e) Reconstructed background. (f) Reconstructed image.

Figure 4.1: Illustration of recovering character image from occlusion.

4.2 Pattern Representation

After noise filtering and binarization procedure, one dimensional representation of an image obtained by using Transformation Ring Projection algorithm. The new representation of an image is translation, scale and rotation invariant, which removes the necessity of selecting invariant features from the extracted feature set. We will give theoretical information about projection algorithm and also represent the outputs for shifted, scaled and rotated character images.

4.2.1 Transformation Ring Projection

The Transformation Ring Projection method proposed by Tang [49] in order to transform two dimensional patterns into one dimension. If the projections are done by using rings, the resulting one dimensional pattern is invariant to rotations.

Definition 4.2.1 *A target pattern set $\mathcal{W}$ denoted as follows:*

$$\mathcal{W} = \{w_1, \ldots, w_i, \ldots, w_N\}, \tag{4.1}$$

where $N \in \mathbb{I}$ and w_i is i th class which may contain finite pattern samples, i.e.,

$$w_i \subseteq \mathcal{A} = \{a_1, \ldots, a_k\} \qquad (i = 1, \ldots, N), \tag{4.2}$$

We call $\mathcal{W}$ a standard set such that

$$|w_i| = 1 \qquad \forall i = 1, \ldots, N \qquad \forall w \in \mathcal{W}, \tag{4.3}$$

That means all w_i in set $\mathcal{W}$ are single pattern sample.

One important aspect of Transformation Ring Projection Method is that it uses a ring extraction panel defined as follows to extract the feature of the pattern samples in order to achieve the orientation invariant property.

Definition 4.2.2 *A ring extraction panel is a triple $\varphi = (\mathcal{R}, \Theta, \delta)$, where $\mathcal{R}, \Theta \in \mathbb{I}$ and δ, ring extraction function, is a function of $\mathcal{R}$ and Θ, i.e., $\delta = p(\mathcal{R}, \Theta)$ and*

$$\forall_{k \in \mathbb{I}} p(\mathcal{R}, \Theta)) = p(\mathcal{R}, \Theta + km), \tag{4.4}$$

That means $p(\mathcal{R}, \Theta)$ is a cycle function of Θ with a period of m.

A graphical representation of Ring-extraction panel is shown in Figure 4.2. It consists of n concentric ring and m spokes, where m and n are integers. The radius of i th ring is denoted by r_i and s_j j th spoke ($i = 1, \ldots, n; \quad j = 1, \ldots, k$). Each cross point between ring and spoke is called sample point and denoted by $p(i, j)$. After going through the ring-extraction panel, a pattern sample is represented by a vector called ring projection vector.

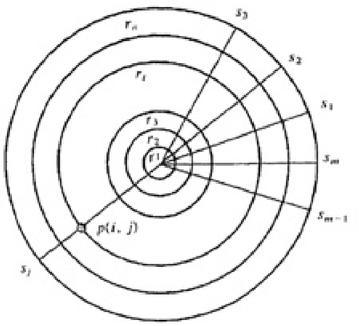


Figure 4.2: Ring extraction panel [49].

Definition 4.2.3 *A ring projection vector extracted by ring-extraction panel* $\varphi = (\mathcal{R}, \Theta, \delta)$ *is*

$$\vec{\mathcal{V}} = \begin{bmatrix} p_{r_1} \\ \vdots \\ p_{r_n} \end{bmatrix} = \begin{bmatrix} \int_1^m p(1, j) dj \\ \vdots \\ \int_1^m p(n, j) dj \end{bmatrix}. \tag{4.5}$$

If we consider the discrete case of the digital image pattern, the ring-projector vector can be represented as

$$\vec{\mathcal{V}} = \begin{bmatrix} \sum_1^m p(1, j) dj \\ \vdots \\ \sum_1^m p(n, j) dj \end{bmatrix}. \tag{4.6}$$

$$p(i, j) = f(x) = \begin{cases} 1, & \textit{if overlaps with pattern sample}, \\ 0, & \textit{otherwise} \; , \end{cases} \tag{4.7}$$

Definition 4.2.4 *The* r_i *is called non-zero ring if* $p_{r_i} \neq 0$ *otherwise* r_i *called zero ring.*

The ring projection operation play important role in Transformation Ring Projection Algorithm. Its basic principle can be seen in Figure 4.3.

Theorem 4.2.5 *Ring projection is a rotation invariant operation, i.e., for a given pattern, the result of the pattern orientation independent [49].*

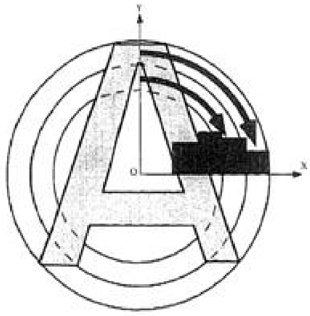


Figure 4.3: Ring projection [49].

Proof. For a given patterns Ω_1 and Ω_2 stands for two samples, between which there is a rotated angle k. The ring projected vectors for Ω_1 and Ω_2 extracted by $\varphi = (\mathcal{R}, \Theta, \delta)$ are

$$\vec{\mathcal{V}}_1 = \begin{bmatrix} p_{r_1^1} \\ \vdots \\ p_{r_n^1} \end{bmatrix}, \quad \vec{\mathcal{V}}_2 = \begin{bmatrix} p_{r_1^2} \\ \vdots \\ p_{r_n^2} \end{bmatrix}, \tag{4.8}$$

For the sample Ω_1 ring projection of i th ring is

$$p_{r_i^1} = \int_1^m p(1, j)dj \qquad (i = 1, \ldots, n)\,, \tag{4.9}$$

and for Ω_1

$$p_{r_i^2} = \int_1^m p(1, j + k)dj \qquad (i = 1, \ldots, n)\,, \tag{4.10}$$

where for certain ring i and k are constant.

Since $p(i, j)$ is a periodic function with period of m it can be written as:

Let $t = j + k$ then we have $dt = dj$

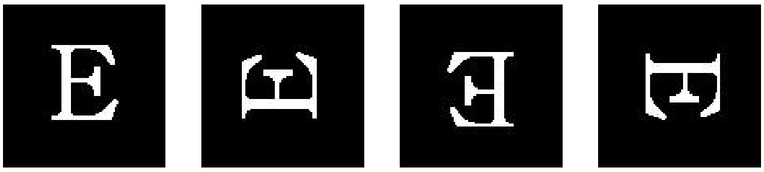
$$p_{r_i^2} = \int_1^m p(1, j + k)dj = \int_k^{m+k} p(1, t)dt, \tag{4.11}$$

$$p_{r_i^2} = \int_k^{m+k} p(1, t)dt = \int_k^m p(1, t)dt + \int_m^{m+k} p(1, t)dt, \tag{4.12}$$

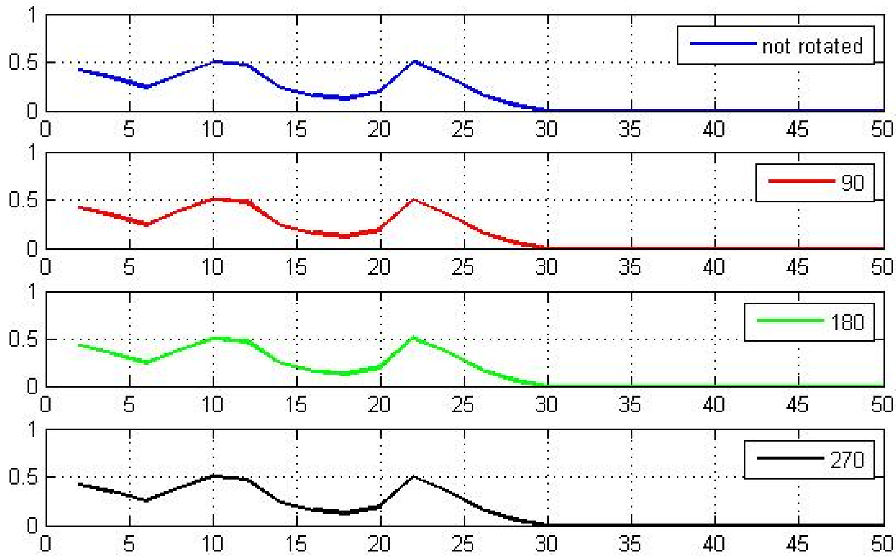
$$p_{r_i^2} = \int_k^m p(1, t)dt + \int_1^k p(1, t)dt = \int_1^m p(1, t)dt = p_{r_i^1}, \tag{4.13}$$

■

The theorems shows that the result of projection, all patterns in the same category but different orientations (i.e., rotated) have same one dimensional pattern. However, ring projection does not solve the size and translation invariance problems. In order to obtain size and translation invariance of one dimensional pattern, it is required to follow some procedures before the ring projection method. We will explain these procedures in the next section.



(a) Not rotated. (b) 90. (c) 180. (d) 270.



(e) Ring Projections.

Figure 4.4: Rotated characters and ring projections.

4.2.2 Position Invariance

Center of the mass is a position and rotation invariant feature for the two dimensional binary image [27]. Let (C_x, C_y) be the center of mass then it can be calculated in following way:

$$C_x = \frac{m_{10}}{m_{00}}, \tag{4.14}$$

$$C_y = \frac{m_{01}}{m_{00}}, \tag{4.15}$$

where m_{pq} is the Cartesian moment of order $p + q$ and is defined as

$$m_{pq} = \sum_x \sum_y x^p y^q p(x, y), \tag{4.16}$$

The position invariance can be obtained by translating the origin of our reference frame to the center of mass [13, 27, 49]. After moving the origin of our reference frame to the center of

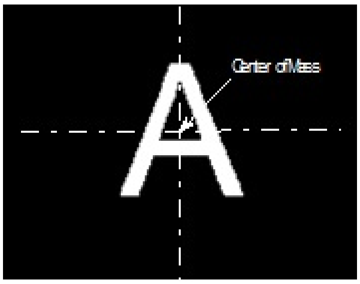


Figure 4.5: Binary image and its center of mass [49].

mass, we let

$$\mathcal{M}(x, y) = \max_{\mathcal{N}\in\mathcal{D}} \left|\mathcal{N}(x, y) - p(C_x, C_y)\right|, \tag{4.17}$$

where $\left|\mathcal{N}(x, y) - p(C_x, C_y)\right|$ represents the Euclidean distance between two points.In addition we transform $p(x, y)$ into polar coordinate based on the following relation:

$$\begin{cases} x = \gamma\cos\theta, \\ y = \gamma\sin\theta, \end{cases} \tag{4.18}$$

Hence,

$$p(x, y) = p(\gamma\cos\theta, \gamma\sin\theta), \tag{4.19}$$

where $\gamma \in [0, \infty)$ and $\theta \in (0, 2\pi]$.For any fixed $\gamma \in [0, M]$, we then compute following integral.

$$f(\gamma) = \int_0^{2\pi} p(\gamma\cos\theta, \gamma\sin\theta)d\theta, \tag{4.20}$$

The resulting $f(\gamma)$ is a ring projection of the planar mass distribution of the two dimensional pattern. However the images are most often stored in discrete formats we must modify $f(\gamma)$ into the following expression:

$$f(\gamma) = \sum_{k=0}^{M} p(\gamma\cos\theta, \gamma\sin\theta)d\theta, \tag{4.21}$$

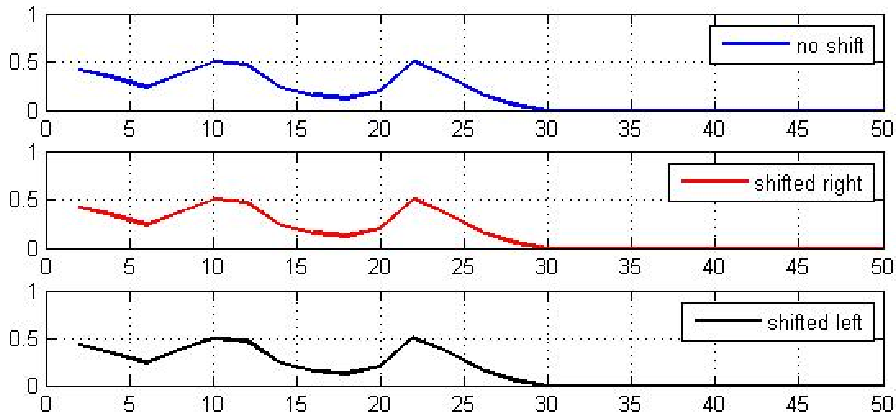
4.2.3 Size Invariance

Size invariance can be obtained by normalizing the size of input pattern to predefined size S. Let us assume s_j is the largest non-zero ring distance of given input pattern. Then the size of

(a) Original image character.

(b) Shifted to right bottom corner.

(c) Shifted to left upper corner.



(d) Ring Projections.

Figure 4.6: Characters and ring projections.

the input pattern is normalized by a factor of $\frac{S}{s_i}$. However, the disadvantage of image resizing is spatial distortion. Its effects on the projections in different scales can be seen at Figure 4.7. Ellipses marks the affect of distortion on output.

4.2.4 TRP Algorithm

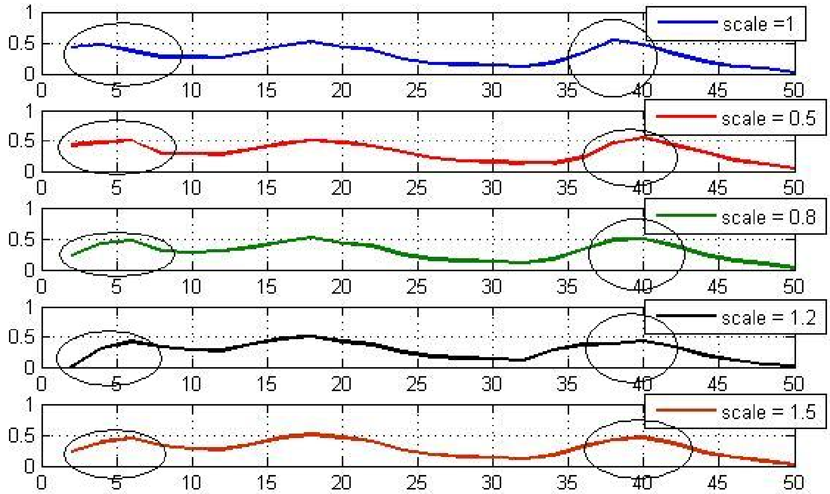
The steps of the Ring Projection Algorithm can be summarized as follows:

1. Normalize the input pattern to standard size.

2. Find center of mass of the two dimensional binary pattern and locate center of reference to the center of mass.

3. Transform normalized pattern to polar coordinate system.

(a) .5. (b) .8. (c) 1. (d) 1.2. (e) 1.5.



(f) Ring Projections.

Figure 4.7: Original and scaled character images and ring projections.

4. Apply ring projection and calculate one dimensional output for increasing ring radius.

4.3 Feature Extraction

In pattern recognition, each pattern for training and test is represented by a feature vector and a discrimination rule is applied to classify a test vector. In our machine printed character recognition approach two different type of feature extraction method were examined, Fourier and Wavelet based. Fourier and Wavelet based feature extraction methods were used for character [13, 26] and many other recognition problems [6, 10, 15, 38, 43, 45, 47, 57]. We have extracted features from one dimensional character representation function obtained in the previous section.

4.3.1 Fourier Analysis

Fourier descriptors were used by Persoon [43], Zahn [57], Chen [13] and Impedovo [26] to describe the characters, numerals and shape of closed planar figures. In this thesis, we have applied Fourier transform after obtaining one dimensional character pattern representation.

Let $y(n)$ is the one dimensional character representation obtained by applying Transformation Ring Projection method, then the coefficients of Discrete Fourier Transform is:

$$\widehat{y}_k = \sum_{n=0}^{N-1} y(n) e^{\frac{-2\pi ikn}{N}} \qquad (k = 0, ..., N-1)\,, \tag{4.22}$$

The low frequency components of the Fourier transform determine global shape and high frequency components determine the details of function. Thus, we can construct the original function using first M coefficients with the inverse Fourier Transform.In addition, elimination of high frequency descriptors is useful for noise cancellation. It is kind of low-pass filtering procedure:

$$y(n) \approx \frac{1}{N} \sum_{n=0}^{M-1} \widehat{y}_n e^{\frac{2\pi ikn}{N}} \qquad (n = 0, ..., N-1) \tag{4.23}$$

Although only M coefficients are used to approximate the function $y(n)$, n still ranges from 0 to $N-1$. In other words, the same number of points is in the approximated function, however not as many terms are used in reconstruction of each point. The reconstruction results with different number of coefficients are shown at Figure 4.9

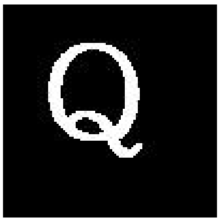
We composed feature vector by considering magnitude spectrum, which provide total amount of information contained at a given frequency, of projected character representation. First 9 Fourier magnitude spectrum values were used as a feature vector in our OCR system implementation. Let F and V denote the frequency spectrum and feature vector respectively:

$$F_i = |F_i|\, e^{j\omega + \phi_i}, \tag{4.24}$$

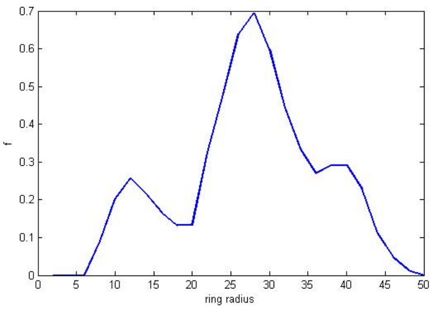
$$\vec{V_1} = \begin{bmatrix} v_1 \\ \vdots \\ v_9 \end{bmatrix} = \begin{bmatrix} |F_1| \\ \vdots \\ |F_9| \end{bmatrix}. \tag{4.25}$$

4.3.2 Wavelet Transform

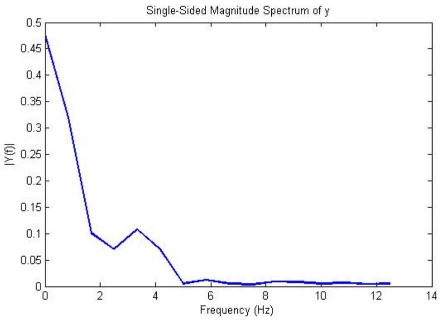
Wavelet transforms are rapidly surfacing in fields as diverse as telecommunication and biology [55]. The main feature of wavelets is that their ability to represent and detect localized fre-



(a) Character image.



(b) Projected character representation (Q).



(c) Single sided magnitude spectrum (Q.)

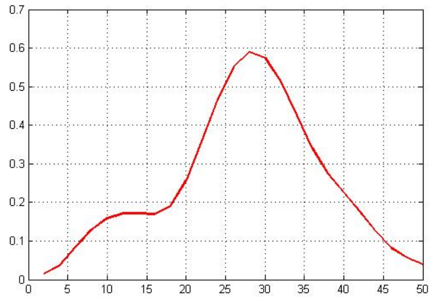
Figure 4.8: Illustration of sample character, its ring projection and single side magnitude spectrum.

quency information about a function or signal. They are suitable for analyzing non-stationary signals where the Fourier transform has difficulties, and become a powerful alternative in many application areas.

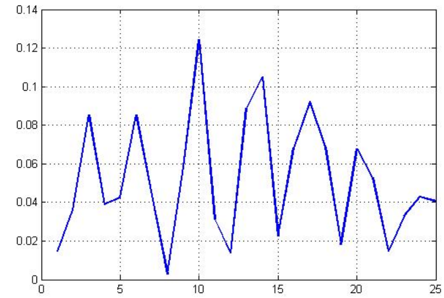
Wavelet transform has been used for character recognition [12, 13, 15, 34, 50], phoneme recognition [35], heart valve sound classification [6]. The main difficulty at wavelet transform based pattern recognition problems is to determine appropriate wavelet basis and number of decomposition level. We have preferred 8 level decomposition to compose feature vector as same length as we did in Fourier transform based feature extraction procedure. In addition, we have find no certain approach determining decomposition level in our literature survey.

There are several approaches have been applied in literature to compose feature vector based on wavelet transform. Combination of maximum,minimum,mean and standard deviation of the wavelet coefficients, energy of signal at each resolution level, and average of absolute value of the signal at each resolution level are some of them.

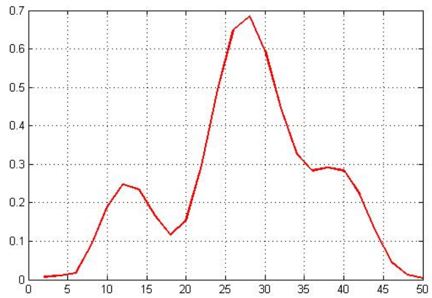
Daubechies 8, Symlet 8 and Coiflet 5 (Figures 3.6,3.7,3.8) wavelets were used to analyze and extract feature vectors. The features are extracted from wavelet decomposition components of the function figure 4.10 using equation 4.26. Thus the each component of signal is expressed



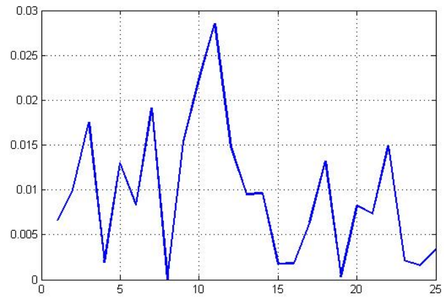
(a) Reconstruction, using only first 3 coefficients.



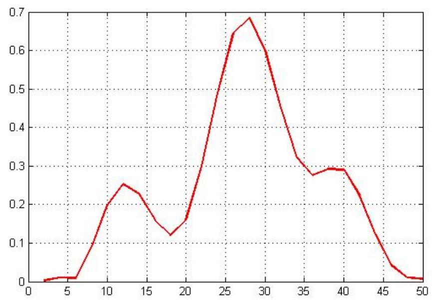
(b) Error.



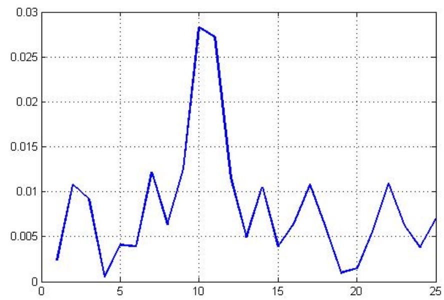
(c) Reconstruction, using only first 5 coefficients.



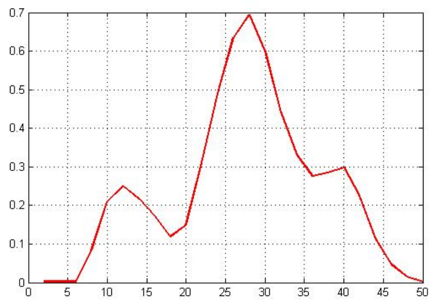
(d) Error.



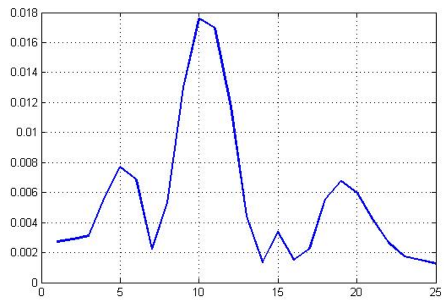
(e) Reconstruction, using only first 7 coefficients.



(f) Error.



(g) Reconstruction, using only first 9 coefficients.



(h) Error.

Figure 4.9: Illustration of reconstruction of function given at figure 4.8.b using first 3,5,7 and 9 Fourier descriptors.

with a single value, where f_j is the average of j-th component of function and the c_{ij} indicates the j th component vector of wavelet decomposition of signal, n is the dimension of the signal in a window.

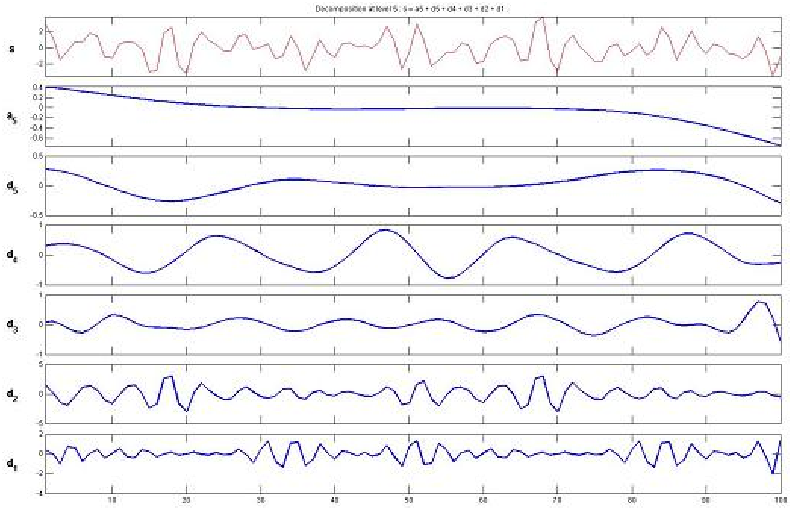


Figure 4.10: The wavelet decomposition.

$$f_j = \frac{\sum_{i=1}^{n} \left| c_{ij} \right|}{n}, \tag{4.26}$$

4.4 Classification Methods

4.4.1 Nearest Neighbor Algorithm

The nearest neighbor algorithm is a method for classifying sample point based on the nearest of a set of previously classified points. It is a type of lazy learning algorithm which means all computations is postponed until classification. The nearest neighbor algorithm is a suboptimal procedure[19]. However it has a probability of a error which is less than twice the Bayes probability and any other decision rule based on the infinite sample set [16, 19].

Assume that we have a new pair (x, ω) and it is desired to estimate class ω by utilizing the information contained in the set of points for which the correct classification is known. We shall call neighbor points $x^{'}$.

$$x^{'} \in \{x_1, \ldots, x_n\}, \tag{4.27}$$

The nearest neighbor to x is defined as

$$\min_{i=1,\ldots,n} d(x_i, x) = d(x_n^{'}, x), \tag{4.28}$$

where d is distance measure. Then decision is made by assigning x to the category of nearest neighbor $x_n^{'}$. The nearest neighbor algorithm allow us to partition the feature space into cells consisting of all points closer to given training point x than to any other training points, and it is called Voronoi tessellation [19].

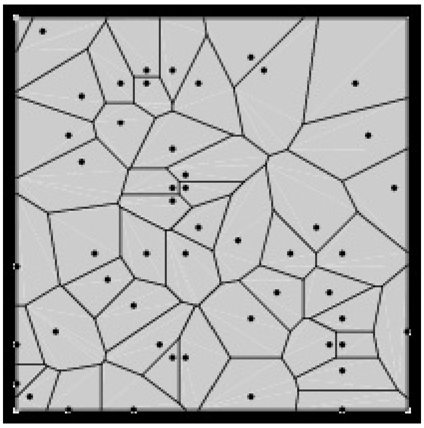


Figure 4.11: Voronoi polygons:The partitioning of a plane with n points into convex polygons such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other.

4.4.2 Linear Discriminant Analysis

Linear discriminant analysis is a well-studied topic in pattern recognition [31]. If underlying densities were unknown, it could be assumed that the parametric form of the discriminant functions were known and it's parameters can be estimated using samples, instead of estimating values or parameters of probability densities. The problem of finding linear discriminant function is formulated as a minimization of some criterion function. Sample risk or training error are the obvious criterion functions.

Main advantage of linear discriminant functions is their simplicity. In some cases it could be desirable to choose implementation simplicity by scarifying classification performance. We have implemented two different linear discriminant method given below.

$$g(x) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_d x_d, \tag{4.29}$$

$$g(x) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_d x_d + \omega_{11} x_1^2 + \omega_{12} x_2^2 + \ldots + \omega_{1d} x_d^2, \tag{4.30}$$

4.4.2.1 Generalized Linear Discriminant Function

Linear discriminant function $g(x)$ can be written as:

$$g(x) = \omega_0 + \sum_{i=1}^{d} \omega_i x_i = \omega_0 + w^T x = a^T y, \tag{4.31}$$

$$y = (1, x_1, \ldots, x_d)^T, \tag{4.32}$$

$$a = (\omega_0, \omega_1, \ldots, \omega_d)^T, \tag{4.33}$$

where ω_i is the weight ω_0 the bias. The discriminant function $g(x)$ give an algebraic measure of the distance from x to hyperplane $\mathcal{H}$. The orientation of $\mathcal{H}$ is determined by the normal vector $\vec{\omega}$ and the location of the surface by the bias ω_0. Polynomial or quadratic discriminant

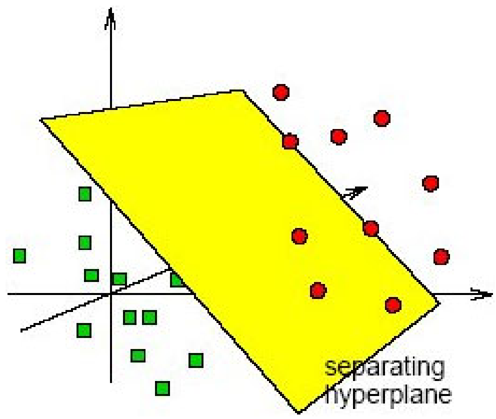


Figure 4.12: Separating hyperplane.

function can be obtained by adding additional terms into equation given above

$$g(x) = \omega_0 + \sum_{i=1}^{d} \omega_i x_i + \sum_{i=1}^{d} \sum_{j=1}^{d} \omega_{ij} x_i x_j = a^T y, \tag{4.34}$$

$$y = \left(1, x_1, \ldots, x_d, x_1^2, x_1 x_2, \ldots, x_d^2\right)^T, \tag{4.35}$$

$$a = (\omega_0, \omega_1, \ldots, \omega_d, \omega_{11}, \omega_{12}, \ldots, \omega_{dd})^T \tag{4.36}$$

Polynomial discriminant function is not linear in x, but it is linear in y. In multicategory case linear discriminant functions divides the feature space into c decision region. Separating hyperplane $\mathcal{H}_{ij}$ is defined by

$$g_i(x) = \omega_{i0} + w_i^T x_i \qquad (i = 1, \ldots, c), \tag{4.37}$$

$$g_i(x) = g_j(x), \tag{4.38}$$

or

$$(w_i - w_j)^T + (\omega_{i0} - \omega_{j0}) = 0, \tag{4.39}$$

where $(w_i - w_j) \perp \mathcal{H}_{ij}$ and distance from x to $\mathcal{H}_{ij}$ is given by $\frac{g_i - g_j}{\|w_i - w_j\|}$. Linear classifiers use the following classification rule:

$$C(x) = C_i \qquad if \quad g_i(x) = \max_{j=1,\ldots,c} g_j(x) \qquad (i = 1, 2, \ldots, c)\,, \tag{4.40}$$

There are several approaches to define criterion function. Perceptron and Minimum Squared Error (MSE) are most popular of them. In our study we concentrated on MSE criterion function. In the next part, we will give information about how MSE formulates and solve criterion function.

4.4.2.2 The Minimum Squared Error Criterion

The MSE formulates the problem as a set of linear equations.

$$a^T y_i = b_i, \qquad b_i > 0 \qquad (i = 1, 2, \ldots, n)\,, \tag{4.41}$$

Let Y be a $n \times (d+1)$ matrix whose i th row is the y_i^T and b the column vector $b = (b_1, \ldots, b_n)^T$ then our problem is to find a weight vector a which minimizes squared error.

$$Ya = b, \tag{4.42}$$

$$Y = \begin{bmatrix} Y_{10} & \ldots & Y_{1d} \\ \vdots & \ddots & \vdots \\ Y_{n0} & \ldots & Y_{nd} \end{bmatrix}, \tag{4.43}$$

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_d \end{bmatrix} \qquad a = \begin{bmatrix} a_1 \\ \vdots \\ a_d \end{bmatrix}, \tag{4.44}$$

$$e = Ya - b. \tag{4.45}$$

Criterion function is defined as follows:

$$J_s = \|Ya - b\|^2 = \sum_i (a^T y_i - b_i)^2. \tag{4.46}$$

Closed form solution of the problem can be found by using gradient

$$\nabla J_s = 2Y^T(Ya - b) = 0, \tag{4.47}$$

$$a = (Y^T Y)^{-1} Y^T b, \tag{4.48}$$

The MSE solution is dependent to the margin vector b. If margin vector is not chosen cautiously MSE doesn't guarantee that the resulting separating hyperplane produce error free classification even if the sample sets are linearly separable. Ho-Kashyap procedure proposes adjustment rules for the margin vector. In the linearly separable case margin vector $\vec{b}$ able to converge to solution. However for the non-separable case inconsistently between minimum squared error and minimum classification error remains [19].

4.4.2.3 Instability Of The Generalized Inverse Solution

Problems are generally defined as ill-posed when a small change in the data may cause a large change in the solution. The solution of equation 4.42 can be extremely unstable when one of the singular values of Y is small. Small singular values cause the generalized inverse solution to be extremely sensitive to small amounts of noise in the data [5].

It is useful to analyze inverse solution from singular value spectrum. Let Y be a m by n matrix. In singular value decomposition matrix Y is factored into:

$$Y = USV^T, \tag{4.49}$$

where

U is m by m orthogonal matrix with columns that are unit basis vector spanning the data space $\mathbb{R}^m$.

V is n by n orthogonal matrix with columns that are unit basis vector spanning the model space $\mathbb{R}^n$.

S is m by n diagonal matrix with nonnegative diagonal elements called singular values.

$$S = \begin{bmatrix} S_p & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.50}$$

S_p is p by p diagonal matrix composed of positive singular values.

$$U = \begin{bmatrix} U_p & U_0 \end{bmatrix}, \tag{4.51}$$

$$V = \begin{bmatrix} V_p & V_0 \end{bmatrix}, \tag{4.52}$$

Expanding of the singular value decomposition representation of Y in terms of U and V gives:

$$Y = USV^T = U_p S_p V_p^T, \tag{4.53}$$

$$(Y^T Y)^{-1} Y^T = V_p S_p U_p^T, \tag{4.54}$$

Generalized inverse solution is given as:

$$a = (Y^T Y)^{-1} Y^T b = V_p S_p U_p^T b = \sum_{i=1}^{p} \frac{U_{.,i} b}{s_i} V_{.,i}, \tag{4.55}$$

The presence of very small singular value s_i in the denominator can thus give us very large coefficients for the corresponding model space $V_{.,i}$ and these basis vector can dominate the solution. Perturbation in the data $b + \Delta b$ is just magnified by factor $1/s_i$ resulting large deviations in the computed solution and become practically useless. The development of theoretical strategies to mitigate this instability is known as regularization theory. One way to stabilize the solution process is to restrict the notion of solution [48]. This stabilized or regularized, the solution in the sense that it made the result less sensitive the data noise [5]. Tikhonov regularization is a technique for regularizing discrete ill-posed problems.

4.4.2.4 Tikhonov Regularization

One of the most popular regularization methods to obtain a meaningful solution is Tikhonov regularization in which the linear system or the least square problem is replaced by the minimization problem. In the simplest case, assume A and B are Hilbert spaces. To obtain regularized solution to equation 4.42, choose a to fit data b in least square sense, but penalize solutions of large norm. Solve minimization problem:

$$a_\alpha = \arg\min_{a \in A} \left\{ ||Ya - b||^2 + \alpha\, ||a||^2 \right\}, \tag{4.56}$$

$\alpha > 0$ is called regularization parameter to be chosen. For some problems (particularly in image restoration) it is better to consider

$$a_\alpha = \arg\min_{a \in A} \left\{ ||Ya - b||^2 + \alpha\, ||La||^2 \right\}, \tag{4.57}$$

where L is typically the discretization of a derivative operator of first or second order. Numerically the problem 4.57 is solved by considering it as a least squares problem.

$$\min \left\| \begin{pmatrix} Y \\ \sqrt{\alpha} L \end{pmatrix} a - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|, \tag{4.58}$$

The solution of equation 4.56 solves the linear system

$$\left(Y^T Y + \alpha I \right) a = Y^T b, \tag{4.59}$$

The choice of the regularization parameter α is crucial since if α is too small the solution is contaminated by the noise in the right-hand side (as it is with $\alpha = 0$, on the other hand if α is too large the solution is a poor approximation of the original problem.

When the dimension is small enough the regularized solution a_α of equation 4.59 can be computed using the singular value decomposition of Y, which is m by n matrix, as:

$$a_\alpha = \sum_{i=1}^{k} \frac{s_i^2}{s_i^2 + \alpha^2} \frac{\left(U_{.,i}\right)^T b}{s_i} V_{.,i}, \tag{4.60}$$

where $k = \min(m, n)$. The quantities

$$f_i = \frac{s_i}{s_i^2 + \alpha^2}, \tag{4.61}$$

are called *filter factor*. For $s_i >> \alpha$, $f_i \approx 1$, and for $s_i << \alpha$, $f_i \approx 0$. For singular values between these two extremes, as the s_i decrease, the f_i decrease monotonically.

If $L = I$ (equation 4.57) the problem is said to be in standard form. Otherwise it is in general form. In this case the generalized singular value decomposition can be used. The factorization of the pair (Y, L) is

$$Y = U \begin{pmatrix} \Gamma & 0 \\ 0 & I_{n-p} \\ 0 & 0 \end{pmatrix} W^{-1}, \qquad L = V \begin{pmatrix} \Upsilon & 0 \end{pmatrix} W^{-1}, \tag{4.62}$$

where L is p by n and $m \geq n \geq p$. The matrices U m by m and V p by p are orthonormal and the nonsingular matrix W is n by n. The matrices Γ and Υ with diagonal elements γ_i and τ_i are diagonal with $\gamma_i + \tau_i = 1$. The generalized singular values of the matrix $\begin{pmatrix} A & L \end{pmatrix}$ are defined as $s_i = \gamma_i / \tau_i$. The last $n - p$ columns of W form a basis of the null space of L. If $L = I$ the *generalized singular value decomposition* reduces to the *singular value decomposition* of Y.

For the generalized form, the regularized solution is written as:

$$a_\alpha = \sum_{i=1}^{p} \frac{s_i^2}{s_i^2 + \alpha^2} \frac{(U_{.,i})^T b}{\gamma_i} W_{.,i} + \sum_{i=p+1}^{n} (U_{.,i})^T b W_{.,i}, \tag{4.63}$$

The second term in the right-hand side is the (unregularized) component of the solution in the null space of L.

Many methods have been devised for choosing α, Morozov's Discrepancy Principle, The L-curve Criterion, Generalized Cross Validation are some of them. We will give brief information about these techniques but detailed information can be found in [5, 23].

Morozov's Discrepancy Principle can be used only if the (norm of the) noise vector ϵ is known [23]. The value of the regularization parameter α is chosen such that the norm of the residual equals the norm of the noise vector using the mathematical solution from equation 4.59,

$$\left\| b - Y \left(Y^T Y + \alpha I \right)^{-1} A^T b \right\| = \|\epsilon\|, \tag{4.64}$$

The L-curve Criterion use curve $(\|a_\alpha\|, \|b - Ya_\alpha\|)$ obtained by varying the value of $\alpha \in [0, \infty)$. This curve is known as the *L-curve* since it is shaped as the letter "L" (Figure 4.13). It is more illuminating to look at this curve in a log-log scale. It is proposed to choose the value α corresponding to the *vertex* or the *corner* of the L-curve that is the point with maximal curvature. A motivation for choosing the *vertex* is, as we said before, to have a balance between α being too small and the solution contaminated by noise, and α being too large giving a poor approximation of the solution [23]. The *vertex* of the L-curve gives an average value between these two extremes.

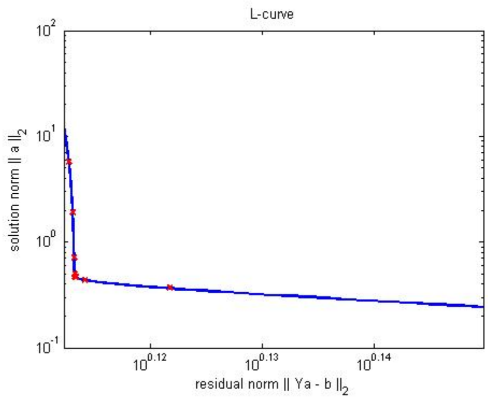


Figure 4.13: L curve.

Generalized cross validation is an alternative method for selecting a regularization parameter α. Cross validation and generalized cross-validation are techniques for model fitting for given data and model evaluation. These two tasks can be accomplished using independent data samples. The available data can be split into two sets, one for fitting and one for evaluation. This is not very efficient if the sample is not very large [23].

Generalized cross validation for large scale linear ill-posed problems uses the techniques of estimation of quadratic forms to compute the parameter of the model. The regularized problem is written as [23]:

$$\min\left\{\|b - Ya\|^2 + m\alpha\,\|a\|^2\right\}, \tag{4.65}$$

where $\alpha > 0$ is the regularization parameter and the matrix Y is m by n. The generalized cross validation estimate of the parameter α is the minimizer of

$$G(\alpha) = \frac{\frac{1}{m}\left\|\left(I - Y\left(Y^T Y + m\alpha I\right)^{-1} Y^T\right)b\right\|^2}{\left(\frac{1}{m}tr\left(I - Y\left(Y^T Y + m\alpha I\right)^{-1} Y^T\right)\right)^2}, \tag{4.66}$$

The numerator is, up to a scaling factor, the square of the norm of the residual corresponding to the solution of the normal equations of the regularized problem [23].

CHAPTER 5

PERFORMANCE EVALUATION

In order to test the efficiency of our character recognition system approach, we have used set of 26 uppercase English characters in 7 different fonts. The original English characters were represented by 100 by 100 pixels. All the reference gray scale character images were prepared by using commercial image softwares and Matlab. Up to this point all the processes can be called segmentation. In our experiment we have used 2 different types of classification method: *Nearest Neighbor Algorithm* and *Linear Discriminant Analysis*. In Linear Discriminant Analysis we have implemented *linear* and *quadratic* Minimum Squared Error discriminant functions as a classification methods. Matlab code for Minimum Squared Error discriminant functions can be found on Appendix A.

This section is divided into 5 subsections. Since translation will not change the relative position of the center of the mass of the character, our major concern was the system performance under rotation, scaling and noise. First we have computed training error for each feature extraction and classification method, because training error gives foreknowledge about efficiency of recognition approach. Next, we have tested our recognition scheme on scaled and rotated characters to observe sensitivity of feature extraction and classification methods under these transformations. Then, performance of each method was examined using combination of randomly rotated and scaled character sets. The last section represents the effect of noise on recognition process. Recognition rates of characters for each test can be found in the next chapter.

5.1 Test Results on Isolated Character Sets

5.1.1 Training Error

First we have observed training errors of each feature extraction and classification methods mentioned above utilizing training data as a test data. Training error gives foreknowledge about efficiency of the given method, but 0% training error does not mean that it will produce 100 % recognition rate. Training error of each classification methods were calculated using 729 training image and recognition rates of each classifier are listed at Table 5.1 for Fourier, Daubechies 8, Symlet 8 and Coiflet 5 wavelets based approximations respectively.

At a first glance to the training errors, we can say that,it is obvious feature space can not be divided by linear discriminant function in our case. Quadratic Mean Squared Error based discriminant function seems to be admissible method even if it has complications recognizing some of the characters. Nearest Neighbor method produced no training error. Features extracted using wavelet and Fourier transforms yielded similar training errors under all classification methods.

Table 5.1: Training error.

Method	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
Fourier	0	26.2363	2.7473
Daubechies 8	0	27.1978	1.5109
Symlet 8	0	26.0989	2.7473
Coiflet 5	0	22.9396	1.511

5.1.2 Experiments on Size Invariance

We have observed the effect of scaling on character recognition process using 11 different scale factor, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5 times of the original character size. Figure 5.1 shows that recognition rates decrease dramatically if the scaling factor get smaller, because of character size normalization procedure. Because, size normalization is causing spatial distortion on character image. The effect of spatial distortion can be seen at figure 5.1. However, while size of scaled image get closer to the predetermined normalized character size, decrease on recognition rates become smaller. In addition experimental results also showed that wavelet based feature sets and linear,quadratic Minimum Squared Error discriminant functions are more sensitive to the distortions compared to Fourier based approach and Nearest Neighbor Algorithm.

Table 5.2: Average recognition rates of scaled characters.

Method	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
Fourier	93.8686	63.3741	80.3946
Daubechies 8	88.9486	63.4615	80.8816
Symlet 8	89.0235	64.5604	81.9056
Coiflet 5	89.3232	66.4835	82.3676

5.1.3 Experiments on Rotation Invariance

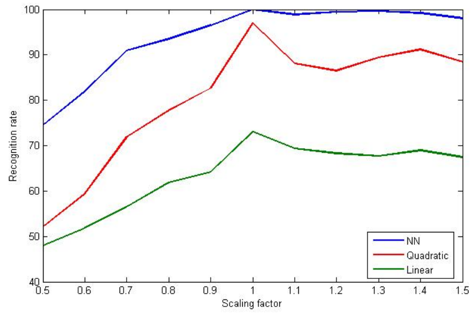
We have used 12 different rotation angle (15,30,45,60,75,90,105,120,135,150,175,180 degrees) in order to monitor the behavior of recognition scheme. 336 images (total 8736) were used for each character. We were expecting that rotation has no or very little influence on recognition rates. However, we observed a small effect for combination of Nearest Neighbor Algorithm and Fourier based feature sets results but there were considerable difference for other combinations (Table 5.3). These results again represented how much sensitive the Mean Squared Error based classifiers and wavelet based feature sets are for our approach. Possible source of change on recognition rate might be distortions or computation errors in image processing steps whilst producing rotated character image.

Table 5.3: Average recognition rates of rotated characters.

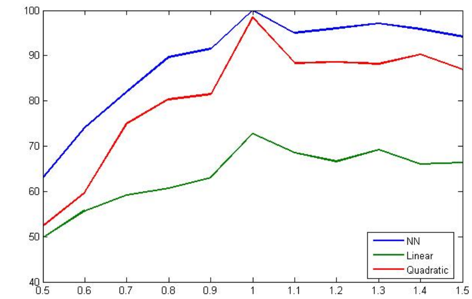
Method	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
Fourier	99.4963	67.2390	89.3773
Daubechies 8	97.1955	68.4982	90.4304
Symlet 8	96.9322	69.5398	89.9611
Coiflet 5	97.4588	72.4817	91.5751

5.1.4 Performance Test

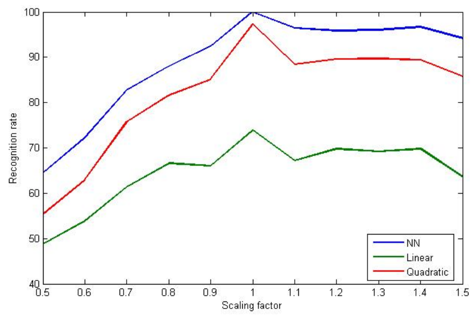
The full performance of each feature extraction and classification methods were tested using 11.648 randomly scaled and rotated character images. Table 5.4 illustrates the performance test results.



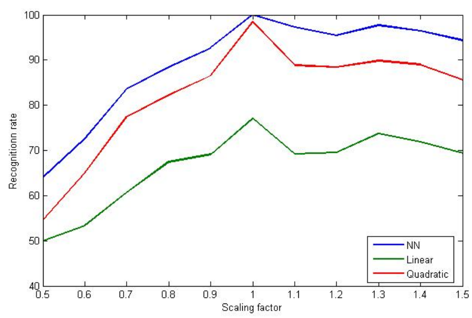
(a) Fourier based.



(b) Daubechies 8 based.

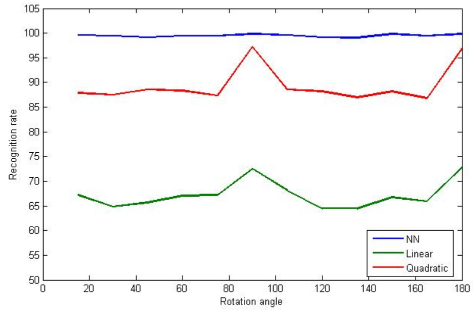


(c) Symlet 8 based.

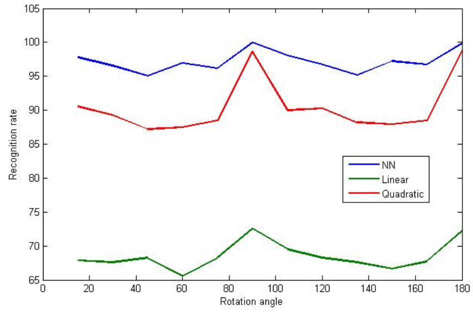


(d) Coiflet 5 based.

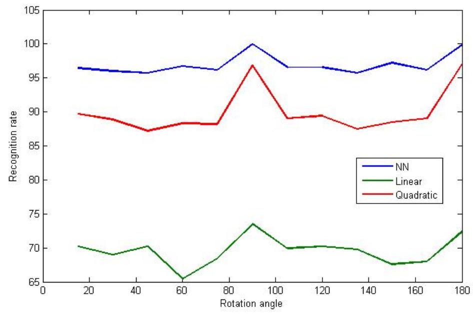
Figure 5.1: Illustration of average recognition rate change for scaled characters.



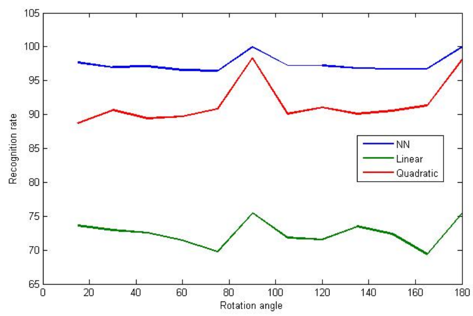
(a) Fourier based.



(b) Daubechies 8 based.



(c) Symlet 8 based.



(d) Coiflet 5 based.

Figure 5.2: Illustration of average recognition rate change for rotated characters.

Table 5.4: Average recognition rates for various scaled and rotated characters.

Method	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
Fourier	96.1367	62.9464	82.3403
Daubechies 8	91.2689	63.6418	82.8039
Symlet 8	90.9942	65.8053	83.6968
Coiflet 5	92.3592	67.4536	84.2119

5.1.5 Effect of Noise

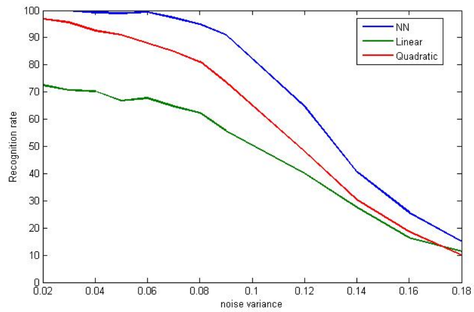
Effect of noise on recognition output were observed using noisy images contaminated by salt and pepper noise having 0.1 mean and variance changing from 0.02 to 0.18. 8736 character image were used in this experiment, sample noisy images can be seen in Figure 5.3. In our

(a) m = 0.1 var = 0.04.

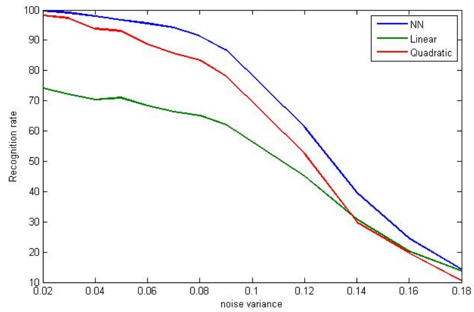
(b) m = 0.1 var = 0.12.

Figure 5.3: Sample noisy images.

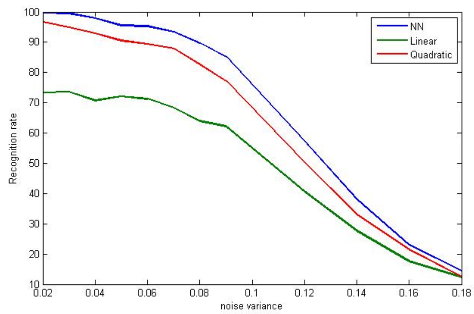
implementation we have applied Median noise filtering implemented by Matlab function *medfilt2* before Transformation Ring Projection procedure. In the first test we did not apply any occlusion removing procedure and have seen that recognition rate reduced dramatically with increasing noise variance (Figure 5.4), because of high noise sensitivity of ring projection method. However, in the second test after noise filtering we have recovered occluded character image by removing foreground occlusion Figure 5.5. Removing foreground occlusion considerably changed recognition rate in the better way.



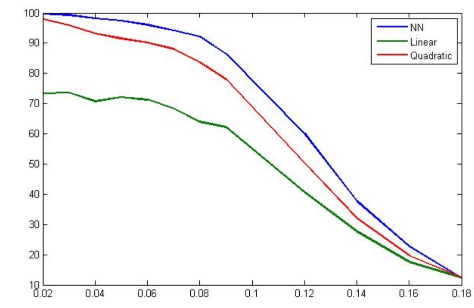
(a) Fourier based.



(b) Daubechies 8 based.

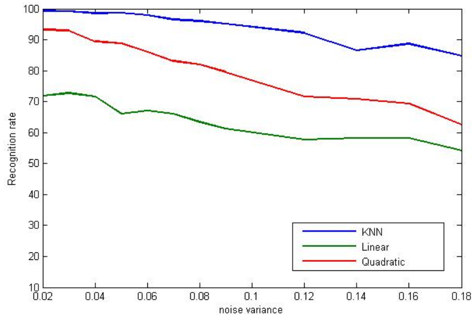


(c) Symlet 8 based.

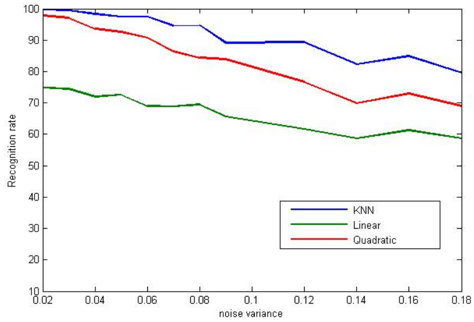


(d) Coiflet 5 based.

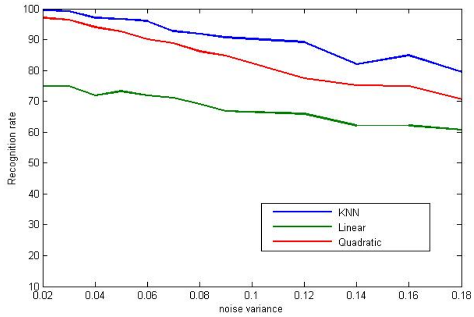
Figure 5.4: Illustration of change in recognition rate with increasing noise variance (without removing occlusion).



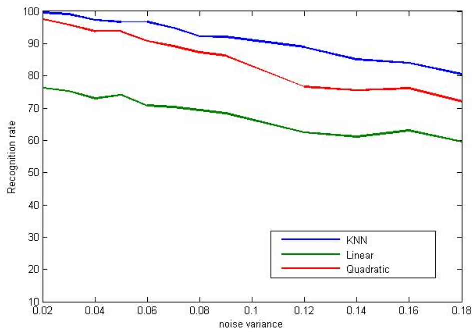
(a) Fourier based.



(b) Daubechies 8 based.



(c) Symlet 8 based.



(d) Coiflet 5 based.

Figure 5.5: Illustration of change in recognition rate with increasing noise variance (with removed occlusion).

CHAPTER 6

EXPERIMENTAL RESULTS

This chapter gives the character fonts which are used in our experiments and also obtained character recognition performances for each isolated character.

6.1 Fonts

(a) TimesNewRoman.

(b) Arial.

(c) Virinda.

(d) Tahoma.

(e) Simsum.

(f) Charming.

(g) KatyBerry.

Figure 6.1: Character fonts.

6.2 Test Results for Scaled Characters

Table 6.1: Average recognition rates for scaled characters (Fourier).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	94.8052	49.3506	69.1558
B	92.5325	69.8052	73.3766
C	96.7532	58.7662	81.4935
D	92.2078	69.4805	84.4146
E	88.9610	36.0390	60.7143
F	90.5844	38.3117	62.9870
G	96.4286	62.3377	70.1299
H	90.9091	58.4416	71.7532
I	88.9610	79.5455	74.3506
J	96.1039	70.1299	87.0130
K	94.8052	65.5844	87.3377
L	94.4805	41.2338	76.2987
M	96.7532	67.2078	72.0779
N	91.2338	53.8961	68.5065
O	98.3766	78.2468	82.4675
P	93.5065	59.0909	72.4026
Q	95.7792	53.2468	84.0909
R	93.1818	47.4026	85.3896
S	95.7792	54.8701	86.0390
T	92.5325	76.9481	88.9610
U	95.1299	74.0260	92.8571
V	99.0260	86.0390	93.5065
W	96.4286	77.9221	95.1299
X	91.5584	67.5325	88.6364
Y	91.5584	69.1558	88.6364
Z	92.2078	83.1169	92.5325
Average	93.8686	63.3741	80.3946

Table 6.2: Average recognition rates of scaled characters (Daubechies 8).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	89.6104	50.3247	63.3117
B	90.9091	69.4805	72.4026
C	89.9351	67.2078	76.6234
D	88.6364	62.3377	77.5974
E	86.3636	27.9221	62.6623
F	78.8961	50.9740	74.0260
G	90.5844	46.7532	75.0000
H	83.7662	65.5844	81.8182
I	77.9221	79.8701	78.5714
J	90.5844	61.6883	76.6234
K	91.5584	67.2078	84.7403
L	90.9091	70.7792	85.7143
M	94.4805	65.2597	82.4675
N	88.3117	51.6234	75.6494
O	79.2208	67.8571	84.7403
P	88.3117	75.0000	78.2468
Q	93.5065	52.2727	80.1948
R	91.2338	50.9740	82.4675
S	81.1688	63.9610	79.8701
T	92.8571	84.4156	85.7143
U	94.4805	68.5065	89.2857
V	99.0260	81.8182	93.5065
W	98.0519	67.5325	93.1818
X	87.3377	62.6623	94.8052
Y	85.0649	61.0390	81.1688
Z	89.9351	76.9481	92.5325
Average	88.9486	63.4615	80.8816

Table 6.3: Average recognition rates of scaled characters (Symlet 8).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	85.3896	53.8961	71.4286
B	88.9610	75.9740	77.2727
C	96.4286	71.1039	82.4675
D	89.9351	75.3247	80.5195
E	83.4416	32.4675	65.2597
F	82.7922	54.5455	61.0390
G	94.4805	61.0390	71.4286
H	85.7143	52.2727	81.1688
I	81.4935	72.7273	77.5974
J	89.2857	52.5974	76.2987
K	87.0130	51.2987	83.1169
L	92.2078	67.5325	87.3377
M	93.8312	62.3377	77.9221
N	82.7922	59.7403	86.6883
O	95.1299	59.0909	77.9221
P	86.3636	69.1558	85.3896
Q	87.9870	66.2338	88.9610
R	89.9351	65.2597	80.8442
S	88.9610	56.4935	86.0390
T	90.9091	82.1429	91.2338
U	91.8831	84.4156	91.8831
V	98.3766	85.0649	96.4286
W	90.9091	70.7792	88.6364
X	87.9870	67.2078	91.8831
Y	83.4416	56.1688	79.2208
Z	88.9610	73.7013	91.5584
Average	89.0235	64.5604	81.9056

Table 6.4: Average recognition rates of scaled characters (Coiflet 5).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	85.7143	54.8701	68.1818
B	94.8052	78.5714	75.6494
C	93.5065	61.3636	85.0649
D	89.6104	55.1948	81.8182
E	79.5455	40.9091	70.4545
F	79.2208	50.0000	64.9351
G	95.7792	66.8831	74.6753
H	84.4156	60.7143	82.7922
I	83.1169	71.4286	91.5584
J	81.8182	64.2857	88.9610
K	87.6623	68.8312	77.2727
L	94.8052	66.2338	89.6104
M	94.1558	57.1429	76.2987
N	89.6104	66.2338	83.4416
O	95.7792	59.7403	80.1948
P	85.7143	72.4026	80.5195
Q	88.6364	70.1299	83.1169
R	81.4935	67.8571	76.9481
S	93.1818	55.8442	82.1429
T	91.5584	74.0260	81.4935
U	89.2857	86.3636	93.5065
V	98.0519	89.6104	94.8052
W	90.2597	82.1429	91.8831
X	94.1558	72.7273	97.0779
Y	88.3117	62.3377	80.5195
Z	92.2078	72.7273	88.6364
Average	89.3232	66.4835	82.3676

6.3 Test Results for Rotated Characters

Table 6.5: Average recognition rates for rotated characters (Fourier).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	99.4048	49.1071	82.1429
B	100	77.6786	85.7143
C	100	55.6548	87.2024
D	98.2143	79.7619	89.5833
E	100	44.9405	83.3333
F	99.1071	48.5119	78.5714
G	99.4048	62.50	81.8452
H	99.7024	68.1548	83.3333
I	95.8333	77.3810	83.6310
J	99.7024	76.1905	93.4524
K	100	74.1071	95.5357
L	100	47.3214	87.5
M	100	75.00	75.5952
N	99.1071	55.0595	81.5476
O	100	79.7619	89.2857
P	100	59.2262	81.25
Q	100	54.1667	95.5357
R	100	52.9762	97.9167
S	99.7024	52.9762	93.4524
T	96.7262	83.3333	95.8333
U	100	78.5714	99.1071
V	100	85.7143	94.6429
W	100	76.7857	97.6190
X	100	73.2143	93.4524
Y	100	73.5119	98.2143
Z	100	86.6071	98.5119
Average	99.4963	67.2390	89.3773

Table 6.6: Average recognition rates of rotated characters (Daubechies 8).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	96.7262	54.4643	73.5119
B	99.4048	73.2143	87.2024
C	96.4286	70.8333	86.3095
D	91.9643	69.6429	89.8810
E	97.9167	33.9286	87.5000
F	91.6667	58.3333	83.6310
G	96.4286	49.7024	87.2024
H	97.3214	66.9643	95.5357
I	88.0952	86.9048	82.1429
J	96.1310	67.8571	87.7976
K	99.4048	72.0238	94.9405
L	100	77.6786	93.1548
M	98.5119	64.8810	97.0238
N	98.2143	53.5714	89.2857
O	92.5595	74.4048	91.3690
P	98.8095	82.7381	94.6429
Q	99.7024	59.5238	86.0119
R	98.5119	59.5238	90.7738
S	95.8333	77.6786	92.8571
T	99.7024	88.3929	89.2857
U	98.5119	67.2619	93.4524
V	100	82.7381	96.7262
W	100	71.4286	95.5357
X	98.5119	70.5357	97.9167
Y	96.7262	62.5000	88.9881
Z	100	84.2262	98.5119
Average	97.1955	68.4982	90.4304

Table 6.7: Average recognition rates of rotated characters (Symlet 8).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	97.3214	55.3571	80.9524
B	93.4524	88.3929	89.2857
C	97.6190	69.0476	86.9048
D	97.3214	83.0357	88.3929
E	98.5119	43.4524	87.5000
F	98.5119	63.0952	72.6190
G	97.9167	60.7143	79.1667
H	92.8571	68.4524	93.7500
I	92.8571	79.4643	81.5476
J	93.7500	52.6786	82.7381
K	96.7262	54.1667	85.7143
L	96.7262	71.1310	89.2857
M	99.4048	65.7738	89.5833
N	97.3214	61.3095	94.6429
O	97.9167	68.7500	88.0952
P	99.4048	72.3214	95.5357
Q	97.3214	70.2381	98.8095
R	97.3214	71.7262	92.2619
S	95.2381	62.7976	97.0238
T	94.9405	85.4167	97.9167
U	96.4286	90.4762	96.7262
V	100	88.3929	96.7262
W	96.1310	75.8929	94.9405
X	99.1071	65.4762	88.9881
Y	97.3214	60.1190	91.9643
Z	98.8095	80.3571	97.9167
Average	96.9322	69.5398	89.9611

Table 6.8: Average recognition rates of rotated characters (Coiflet 5).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	96.1310	55.0595	80.6548
B	99.4048	88.9881	83.6310
C	96.7262	66.0714	88.6905
D	99.7024	66.6667	90.7738
E	97.0238	52.3810	88.3929
F	92.8571	56.5476	80.3571
G	100	77.9762	85.7143
H	98.5119	70.5357	96.4286
I	93.7500	82.1429	95.5357
J	92.5595	64.5833	92.8571
K	93.1548	72.0238	85.4167
L	99.7024	63.9881	92.2619
M	99.4048	60.1190	91.0714
N	98.8095	77.0833	96.4286
O	98.2143	65.1786	97.9167
P	95.5357	69.9405	92.2619
Q	99.7024	83.0357	96.4286
R	95.2381	73.5119	85.1190
S	99.4048	58.9286	93.7500
T	97.3214	80.6548	88.9881
U	95.8333	96.4286	97.9167
V	99.1071	95.8333	98.2143
W	96.7262	88.0952	96.7262
X	100	65.4762	98.2143
Y	99.1071	75.0000	90.7738
Z	100	78.2738	96.4286
Average	97.4588	72.4817	91.5751

6.4 Performance Test Results

Table 6.9: Performance test results (Fourier).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	96.6518	45.7589	74.3304
B	96.875	73.8839	74.3304
C	97.5446	51.7857	77.0089
D	94.1964	69.8661	85.0446
E	94.8661	32.5893	67.4107
F	92.1875	36.3839	64.2857
G	97.9911	57.8125	73.4375
H	94.8661	61.6071	72.3214
I	89.7321	80.5804	75.6696
J	99.1071	70.0893	88.8393
K	96.2054	63.6161	89.7321
L	99.1071	43.0804	80.1339
M	97.0982	64.9554	64.5089
N	96.6518	54.2411	75
O	99.3304	77.2321	83.0357
P	97.0982	56.0268	78.5714
Q	97.5446	51.5625	85.9375
R	95.3125	48.8839	90.6250
S	96.875	52.4554	89.5089
T	93.75	77.2321	89.2857
U	97.9911	75.2232	93.0804
V	98.4375	88.8393	94.4196
W	98.2143	77.9018	94.8661
X	92.8571	69.4196	92.4107
Y	92.8571	70.0893	92.1875
Z	96.2054	85.4911	94.8661
Average	96.1367	62.9464	82.3403

Table 6.10: Performance test results (Daubechies 8).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	93.0804	45.9821	62.7232
B	95.3125	69.8661	77.2321
C	93.3036	66.7411	75.6696
D	89.9554	58.0357	77.9018
E	93.0804	29.4643	74.3304
F	78.3482	51.3393	71.4286
G	91.0714	47.5446	75.4464
H	89.5089	63.8393	84.1518
I	75.8929	76.5625	81.6964
J	93.9732	64.0625	86.8304
K	92.8571	68.5268	86.1607
L	96.8750	71.6518	85.2679
M	94.6429	66.9643	85.2679
N	91.5179	59.1518	80.5804
O	79.0179	68.3036	85.7143
P	91.5179	68.75	82.5893
Q	95.7589	57.8125	82.8125
R	93.7500	54.9107	82.5893
S	85.0446	66.0714	81.6964
T	94.4196	84.5982	82.8125
U	94.4196	64.5089	90.8482
V	98.4375	78.3482	93.75
W	97.5446	71.875	93.0804
X	89.9554	60.0446	91.5179
Y	89.9554	58.2589	84.5982
Z	93.75	81.4732	96.2054
Average	91.2689	63.6418	82.8039

Table 6.11: Performance test results (Symlet 8).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	87.9464	47.3214	66.9643
B	90.4018	78.5714	79.4643
C	97.3214	66.9643	80.8036
D	91.2946	74.7768	81.9196
E	89.5089	41.0714	75.6696
F	84.5982	54.0179	65.4018
G	93.75	61.3839	70.0893
H	87.5	56.6964	85.7143
I	81.6964	77.6786	80.5804
J	94.4196	52.4554	80.5804
K	86.1607	54.6875	83.2589
L	95.5357	68.3036	89.7321
M	94.8661	65.4018	77.6786
N	91.2946	61.1607	93.9732
O	93.9732	63.3929	77.4554
P	91.0714	71.2054	87.2768
Q	90.4018	65.1786	93.0804
R	89.5089	67.8571	82.5893
S	89.0625	59.8214	87.2768
T	91.7411	82.5893	91.7411
U	95.3125	86.1607	93.75
V	97.5446	87.7232	97.5446
W	92.6339	72.5446	90.1786
X	89.9554	61.3839	86.8304
Y	86.3839	56.6964	82.3661
Z	91.9643	75.8929	94.1964
Average	90.9942	65.8053	83.6968

Table 6.12: Performance test results (Coiflet 5).

Character	Nearest Neighbor(%)	MSE Linear(%)	MSE Quadratic(%)
A	87.7232	45.3125	65.1786
B	97.5446	84.3750	78.5714
C	97.0982	56.6964	83.7054
D	93.0804	62.2768	85.4911
E	89.9554	41.7411	78.7946
F	78.7946	50.6696	65.4018
G	97.9911	67.8571	77.2321
H	89.9554	62.5	86.6071
I	88.3929	73.2143	92.4107
J	86.8304	64.7321	90.4018
K	86.8304	68.3036	72.5446
L	96.875	63.3929	90.6250
M	97.5446	53.5714	80.8036
N	93.9732	70.7589	90.4018
O	96.2054	65.4018	87.7232
P	87.5	71.4286	85.7143
Q	94.8661	73.8839	89.0625
R	83.7054	67.4107	77.4554
S	96.2054	57.1429	82.8125
T	92.8571	77.6786	81.2500
U	93.5268	90.8482	94.1964
V	98.2143	93.3036	93.9732
W	91.7411	83.9286	91.9643
X	96.2054	68.5268	95.7589
Y	91.0714	62.9464	79.6875
Z	96.6518	75.8929	91.7411
Average	92.3592	67.4536	84.2119

CHAPTER 7

CONCLUSIONS

In this thesis we studied rotation and size invariant character recognition system with Fourier and wavelet based feature extraction methods and compared the results. It was observed that Linear Mean Squared Error classifier is not suitable for our character recognition approach.

Quadratic Mean Squared Error classifier has performed better result compared to linear companion, but not as good as Nearest Neighbor classifier. Nevertheless, we can conclude that higher order classifier may produce better results but, determination of order of the separating polynomial is another problem to be solved. Another point noticed our attention is, wavelet based feature set and Mean Squared Error classifiers has preferred compliance compared to Fourier based feature set and Mean Squared Error classifier combination.

The Fourier feature sets and Nearest Neighbor classifier were showed the best performance. Approximately 96% recognition rate were observed for characters in various size and orientation.

Daubechies, Symlet and Coiflet wavelets were used for feature extraction. Each of them produced similar recognition rate between 91% - 92.3% , but Coiflet were the first rate. There are several wavelet based feature extraction method reported in literature for different applications. We have used average value of function in each resolution level. Other methods may worth to try in order to increase the recognition performance.

Noise sensitivity is the weakest point of implemented character recognition system. We observed that recognition rate degraded dramatically with increasing noise variance even if we have used Matlab implementation of median filter. This is because sensitivity of Transformation Ring Projection Algorithm to the foreground occlusion. However, foreground occlusion removing algorithm alleviated this problem considerably and we have obtained better recognition rate for images contaminated by salt and pepper noise.

REFERENCES

[1] R. Abdelhamid, *Reconstruction of Objects from Images with partial Occlusion*, Dissertation zur Erlangung des Grades Doktor der Naturwissenschaften, an der Fakultat für Physik und Astronomie, der Ruhr-Universitat Bochum, 2005.

[2] D. Addison, S. Wermter, G. Arevian, *A Comparison of Feature Extraction and Selection Techniques*, Proceedings of the International Conference on Artificial Neural Networks, Istanbul, Turkey, pp. 212-215, June 2003.

[3] N. Arıca, *An Offline Character Recognition System for Free Style Handwritting*, Ms. thesis, The Graduate School of Natural and Applied Science of the Middle East Technical University September, 1998.

[4] N. Arıca, F.T. Yarman-Vural, *An Overview of Character Recognition Focused on Off-line Hadwriting*, IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, Vol. 31, No.2, pp.216-232, 2001.

[5] R.C. Aster, B. Borchers, C.H. Thurber, *Parameter Estimation and Inverse Problems*, Elsevier Academic Press, 2005, ISBN: 0-12-065604-3.

[6] P.M. Bentley, P.M. Grant and J.T.E. McDonnell, *Time Frequency and Time Scale Techniques for the Classification of Native and Bioprosthetic Heart valve Sounds*, IEEE Transaction on Biomedical Engineering, Vol. 45, No. 1, January 1998, pp. 125-128.

[7] A. Boggess, F.J. Narcowich, *First Course in Wavelets with Fourier Analysis*, Prentice Hall, 1st edition (January 16, 2001), ISBN-10: 0130228095.

[8] R.N. Bracewell, *The Fourier Transform*, Scientific American, June 1989, pp. 86-95.

[9] A. Broersma, *Modular Neural Network Classifier for Optical Occluded Character Recognition*, Ms. thesis, University of Groningen, Faculty of Mathematics and Natural Sciences, 2001.

[10] J. Cai, Z.Q. Lui, *Hidden Markov Model with Spectral Features for 2D Shape Recognition*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.23, No.12, December 2001.

[11] Q. Chen, *Evaluation of OCR Algorithms for Images with Different Spatial Resolutions and Noises*, Ms. thesis, School of Information Technology and Engineering Faculty of Engineering University of Ottawa, 2003.

[12] G.Y. Chen, T. D. Bui, A. Krzyak, *Invariant pattern recognition using radon, dual-tree complex wavelet and Fourier transforms*, Pattern Recognition Journal, Volume 42, Issue 9, September, 2009.

[13] G. Chen, T.D. Bui, *Invariant Fourier-wavelet descriptor for pattern recognition*, Elsevier Science Ltd., 1999, Pattern Recognition, Vol. 32, pp. 1083-1088.

[14] A. Choksuriwong, B. Emile, H. Laurent, C. Rosenberger, *Comparative study of global invariant descriptors for object recognition*, Journal of Electronic Imaging, 17(02), 023015, 1 April 2008.

[15] G.C.H. Chuang and C.C.J. Kuo, *Wavelet Descriptors of Planar Curves:Theory and Applications*, IEEE Transaction on Image Processing, Vol. 5, No. 1, January 1996, pp. 56-70.

[16] T.M. Cover, P.E. Hart, *Nearest Neighbor Pattern Classification*, IEEE Transaction on Information Theory, Vol. IT-13, No. 1, (January 1967), pp. 21-27.

[17] D. Cvetkovic, E.D. Übeyli, T. Cosic, *Wavelet transform feature extraction from human PPG, ECG and EEG signal responses to ELF PEMF exposures: A pilot study*, Digital Signal Processing, 18(5), pp. 861-874, 2008.

[18] S. Doguscu, M. Ünel, *Character Recognition Using Canonical Invariants*, Lecture Notes in Computer Science, Heidelberg : Springer Berlin, 2004, Vol. 3211/2004, [ed.] A. Campilho and M. Kamel.

[19] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification 2nd Edition*, Wiley-Interscience, (October 2000), ISBN-10: 0471056693.

[20] J. Flusser, T. Suk, *Affine moment invariants: A new tool for character recognition*, Pattern Recognition, **15**, (1994), pp. 433-436.

[21] C.O. de A. Freitas, L.S. Oliveira, F. Bortolozzi, *Zoning and metaclasses for character recognition*, Symposium on Applied Computing Proceedings of the 2007 ACM symposium on Applied computing Seoul, Korea 2007, SBN:1-59593-480-4, pp. 632 - 636.

[22] B. Gatos, N. Papamarkos, C. Chamzas, *A Binary Tree Based OCR Technique for Machine Printed Characters*, Engineering Applications of Artificial Intelligence, Volume 10, Issue 4, Pages 403-412, August 1997.

[23] G.H. Golub, G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton Series in Applied Mathematics, Princeton University Press, 41 William Street, Princeton, New Jersey 08540, 2010, ISBN: 978-0-691-14341-5.

[24] I. Guyon, A. Elisseff, *An Introduction to Variable and Feature Selection*, Lecture Notes in Computer Science. Heidelberg : Springer Berlin, 2004, Vol. 3211/2004, [ed.] A. Campilho and M. Kamel.

[25] C. Heil, D.F. Walnut, *Fundamental Papers in Wavelet Theory*, Princeton University Press, 41 William Street Princeton, New Jersey 08540, ISBN: 0-691-11453-6.

[26] S. Impedovo, B. Marangelli, A.M. Fanelli, *A Fourier Descriptors Set for Recognizing Nonstylized Numerals*, IEEE Transaction on Systems Man and Cybernetics, Vol.SMC-8, No. 8, August 1978.

[27] A. Iqbal, A.B.M. Musa, Md. A. Sattar, MD. M. Islam, K. Murase, *A novel algorithm for Translation, Rotation and Scale invariant character recognition*, International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on advanced Intelligent Systems (SCIS and ISIS 2008), Nagoya, Japan, (2008).

[28] A.K. Jain, *Fundamentals of Image Processing*, Prentice Hall, Englewood Cliffs, NJ 07632, (1989), ISBN: 0-13-336165-9.

[29] A.K. Jain, R.P.W. Duin, J. Mao, *Statistical Pattern Recognition: AReview*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, **1**, (2000), pp. 4-37.

[30] M.I. Jordan, C.M. Bishop , *Neural Networks*, ACM Computing Surveys, Vol.28, No.1, March 1996.

[31] B.H. Juang, S. Katagiri, *Discriminative Learning for Minimum Error Classification*, IEEE Transaction on Signal Processing, Vol. 40, No. 12, (December 1992), pp. 3043-3054.

[32] M. Kavianifar, A. Amin, *Preprocessing and Structural Features for a Multi-Fonts Arabic/Persian OCR*, Fifth International Conference on Document Analysis and Recognition (ICDAR'99), ISBN: 0-7695-0318-7. Bangalore, India, (1999).

[33] A. Khotanzad, Y.H. Hong, *Rotation Invariant Pattern Reconition Using Zernike Moments*, 9th International Conference on Pattern Recognition, Rome, Italy, (1988), Vol. 1, pp. 326 - 328, ISBN: 0-8186-0878-1.

[34] S.W. Lee, Y.J. Kim, *Multiresolution recognition of handwritten numerals with wavelet transform and multilayer cluster neural network*, Third International Conference on Document Analysis and Recognition (ICDAR'95), Volume 2, Montreal, Canada, August 14-15, ISBN: 0-8186-7128-9.

[35] C.J. Long, S. Datta, *Wavelet Based Feature Extraction for Phoneme Recognition*, Proc. of 4th Int. Conf. of Spoken Language Processing ICSLP96, 1996, Vol. 1, October 3-6, Philadelphia, PA, USA.

[36] S.G. Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, July 1989, pp.674-693.

[37] S.G. Mallat, *Multiresolution Approximations and Wavelet Orthonormal Bases of L2(R)*, Transactions of the American Mathematical Society, Volume 315, Number 1, September 1989, pp.69-87.

[38] Y. Mallet, D. Coomans, J.K. and O.D. Vel, *Classification Using Adaptive Wavelets for Feature Extraction*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 10, October 1997, pp. 1058-1066.

[39] E.L. Menu, Y.L. Lecourtier, *A new method of distortion-invariant character recognition*, 11th IAPR International Conference on Pattern Recognition Methodology and Systems Proceedings, Vol.2 1992, pp. 599 - 602, ISBN: 0-8186-2915-0.

[40] M.S. Mori, M.J. Yamato, *Robust character recognition using adaptive feature extraction*, Image and Vision Computing, New Zealand IVCNZ, 2008, 23rd International Conference, Christchurch, New Zealand, (2008), pp. 1-6, ISBN:978-1-4244-3780-1.

[41] S. Mori, C.Y. Suen, K. Yamamoto, *Historical Review of OCR Research and Development*, Proceeding of The IEEE, Vol. 80, No. 7, July 1992, pp. 1029-1058.

[42] S. Ozawa, M. Kotani, *A Study of Feature Extraction and Selection Using Independent Component Analysis*, Proc. of Int. Conf. on Neural Information Processing 2000 (ICONIP2000-Taejon, Korea), Vol. 1, pp. 369-374.

[43] E. Persoon, K.S. Fu, *Shape Discrimination Using Fourier Descriptors*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 3, May 1986, pp. 388-397.

[44] F. Pereira, T. Mitchell, M. Botvinick, *Machine learning classifiers and fMRI: A tutorial overview*, NeuroImage, Vol. 45, No. 1, (21 March 2009), pp. 199-209.

[45] S. Pittner and S.V. Kamarthi, *Feature Extraction From Wavelet Coefficients for Pattern Recognition Tasks*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 21, No. 1, 1999, pp. 83-88.

[46] R. Ramanathan, K.P. Soman, *Robust Feature Extraction Technique for Optical Character Recognition*, International Conference on Advances in Computing, Control, and Telecommunication Technologies, (2009), pp. 573-575.

[47] O.A. Rosso, M.T. Martin, A. Plastino, *Brain Electrical Activity Anaysis Using Wavelet Based Informational Tools*, Physica A, 313, (2002), pp. 587-608.

[48] O. Scherzer, *Handbook of Mathematical Methods in Imaging*, Springer Science+BusinessMedia, LLC 2011, ISBN: 978-0-387-92920-0.

[49] Y.Y. Tang, H.D. Cheng, C.Y. Suen, *Transformation-ring projection (TRP) algorithm and its VLSI implementation*, International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), Vol. 5, Issue **1-2**, (1991), pp. 25-56.

[50] Y.Y. Tang, B.F. Li, H. Ma, J. Liu, C.H. Leung, C.Y. Suen, *A Novel Approach to Optical Character Recognition Based on Ring- Projection-Wavelet- Fractal Signatures.*, IEEE Computer Society, Vol. 2, Washington DC, USA, SBN ISSN:1051-4651, 0-8186-7282-X, (1996), pp. 325-329.

[51] C.C. Tappert, C.Y. Suen and T. Wakahara, *The State of the Art in On-line Handwirting Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 8, August 1990, pp 787-808.

[52] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Academic Press, (1999), ISBN: 0-12-686140-4.

[53] L.A. Torres-Mendez, J.C. Ruiz-Suares, L.E. Sucar, G. Gomez, *Translation, Rotation and Scale-Invariant Object Recognition*, IEEE Transaction on Systems, Man, and Cybernetics Part C: Applications and Reviews, Vol. 30, No:1, (2000), pp. 125-130.

[54] D. Trier, A. K. Jain, T. Taxt, *Feature Extraction Methods for Character Recognition- A Survey*, Pattern Recognition, Vol. 29, **4**, (1996), pp. 641-662.

[55] İ. Türkoğlu and A. Arslan, *An intelligent pattern recognition system based on neural network and wavelet decomposition for interpretation of heart sounds*, 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC2001), Vol. 2, pp. 1747-1750, İstanbul 2001.

[56] A.R. Webb, *Statistical Pattern Recognition*, John Wiley & Sons, Ltd. Second Edition, 2002, ISBN: 0-470-84513-9.

[57] C.T. Zahn, R.Z. Roskies, *Fourier Descriptors for Plane Closed Curves*, IEEE Transaction on Computers, Vol. C-21, No.3, (March 1972), pp. 269-281.

APPENDIX A

SOURCE CODE

A.1 Center of Mass

```
%////////////////////////////////////////////////////////////
% Author: Önder Nazım Onak
% data2D: 2 dimensional data f(x,y)
% p: order of the variable x
% q: order of the variable y
% mpq: (p+q) order cartesian moment of the f(x,y)
%This function calculates the cartesian moment of order (p+q) of the
%two dimensional input data
%////////////////////////////////////////////////////////////
function [ mpq ] = CartesianMoment2D( data2D ,p,q)
mpq= 0;
[m n] = size(data2D);
    for i = 1:m
        for j = 1:n
            mpq = mpq + (i^p)*(j^q)*data2D(i,j);
        end
    end
end

%////////////////////////////////////////////////////////////
% Author: Önder Nazım Onak
% input: 2 dimensional binary data.
```

```
% x0: integer x coordinate of centre of mass
% y0: integer y coordinate of centre of mass
%////////////////////////////////////////////////////////////
function [ x0,y0 ] = CenterofMass2D( input )
  m10 = CartesianMoment2D(input,1,0);
  m01 = CartesianMoment2D(input,0,1);
  m00 = CartesianMoment2D(input,0,0);
  x0 = round(m10/m00);
  y0 = round(m01/m00);
end
```

A.2 Transformation Ring Projection

```
function overlay2 = RemoveOcclusion(I_cropped)

I_cropped = medfilt2(I_cropped,[3 3]);
I_eq = adapthisteq(I_cropped);
bw = im2bw(I_eq, graythresh(I_eq));
bw2 = imfill(bw,'holes');
bw3 = imopen(bw2, ones(5,5));
bw4 = bwareaopen(bw3, 40);
bw4_perim = bwperim(bw4);
overlay1 = imoverlay(I_eq, bw4_perim, [256 256 256]);
mask_em = imextendedmax(I_eq, 80);
mask_em = imclose(mask_em, ones(2,2));
mask_em = imfill(mask_em, 'holes');
mask_em = bwareaopen(mask_em, 20);
overlay2 = imoverlay(I_eq, ~(bw4_perim | mask_em), [0 0 0]);
end

%////////////////////////////////////////////////////////////
```

```
% Author: Önder Nazım Onak
% Returns biggest ring radius for the Ring Projection Method
% radius:
% binaryData: 2 dimensional binary data
%///////////////////////////////////////////////////////////////
function [ radius ] = RPLRadius( binaryData )
radius = 0;
[ x0,y0 ] = CenterofMass2D( binaryData );
[m n] = size(binaryData);
    for i = 2:m-1
        for j = 2:n-1
            if(binaryData(i,j) == 1)
                if(binaryData(i-1,j) == 1)
                    if(binaryData(i,j-1) == 1)
                        if(binaryData(i-1,j-1) == 1)
                            distance = sqrt((i - x0)^2 + (j - y0)^2);
                            if(distance >= radius)
                                radius = ceil(distance);
                            end
                        end
                    end
                end
            end
        end
    end
 end

%////////////////////////////////////////////////////////////
%Author: Önder Nazım Onak
% image: binary image full path
% fr: ring projected output of binary image
%////////////////////////////////////////////////////////////
function [ fr ringRadius] = RingProjection(image)
```

```
standartSize = 50;
RadiusStepSize = 2;
AngleStepSize = pi/72;

%remove noise
gscaleim = imread(image);
gscaleim = medfilt2(gscaleim,[3 3]);
gscaleim = Occlusion(gscaleim);
% Normalize input image and convert to binary
binaryImage = im2bw(gscaleim);
[ radius ] = RPLRadius( binaryImage );
normalizedData = imresize(gscaleim, standartSize/radius,'bicubic');
normalizedData = im2bw(normalizedData);

[xnor ynor] = size(normalizedData);
% Find Center of mass
[ x0,y0 ] = CenterofMass2D( normalizedData );

    fr = zeros(standartSize/RadiusStepSize,1);
    ringRadius = zeros(standartSize/RadiusStepSize,1);
    index = 1;

    for r = RadiusStepSize:RadiusStepSize:standartSize
        for i = 0:AngleStepSize:2*pi
            x = round(x0 + r*cos(i));
            y = round(y0 + r*sin(i));
            if((x > 0)&(y > 0) )
                if((x < xnor)&&(y < ynor) )
                    fr(index) = fr(index) + normalizedData(x,y);
                else
                    fr(index) = fr(index);
                end
            else
```

```
                fr(index) = fr(index);
            end
        end
        fr(index) = fr(index)/(2*pi/AngleStepSize);
        ringRadius(index) = r;
        index = index + 1;
    end
end
```

A.3 Feature Extraction

```
%//////////////////////////////////////////////////////////
% Author: Önder Nazım Onak
% Compute Magnitude and phase spectrum of TRP projected pattern
%/////////////////////////////////////////////////////////////
function [magnitude freq] = ComputeTRPFrequecySpectrum(y,r)
L=length(y);

%sampling frequency
Fs = L;
% Calculate the maximum frequency that can be perceived
Fnyquist = Fs/2;

% Since FFT only applied for data when number of point is power of 2
% set length of fft point to next power of 2 from length of data
NFFT = 2^nextpow2(L);

%Apply Discrete Fourier tarnsform
Y = fft(y,NFFT)/L;

% take only positive frequency half of data
Y = Y(1:NFFT/2);
```

```
% calculate power spectrum and phase
magnitude = 2*abs(Y);

% calculate corresponding frequencies for Y
freq = Fs/2*linspace(0,1,NFFT/2);
end


%//////////////////////////////////////////////////////////
% Author: Önder Nazım Onak
%/////////////////////////////////////////////////////////////
function featureVector = ComputeTRPWaveletSpectrum(y,r,type,level)
[c,l] = wavedec(y,level,type);
featureVector = zeros(1,level + 1);

ax = wrcoef('a',c,l,type,level);

featureVector(1) = mean(abs(ax));
% Reconstruct detail coefficients at levels 1:level
% from the wavelet decomposition structure [c,l]

for ii = 1: level
dox{ii} = wrcoef('d', c, l, type, ii);
featureVector(level + 2 - ii) = mean(abs(dox{ii}));
end;
end
```

A.4 Classifier Implementation

```
%//////////////////////////////////////////////////////////
% Author: Önder Nazım Onak
```

```
% Estimate the vector w normal to the least square linear discriminant
% hyperplane
% data is the matrix: each row is one data point, each column is one
% feature (dimension) of the data
%returns the weights of the hyperplanes.
%///////////////////////////////////////////////////////
function [w] = msedTrain(data,group,type)

  noc = max(group);
  for i =1:noc
    classes(i) = i;
    ci = find(group == i);
    classdata(i,:,:) = data(ci,:);
  end

  for i = 1:noc
    for j = (i+1):noc
      [xdimi ydimi zdimi] = size(classdata(i,:,:));
      [xdimj ydimj zdimj] = size(classdata(j,:,:));
      b = ones(ydimi + ydimj,1);
      bmin = 0.1*ones(ydimi + ydimj,1);

      if(strcmp(type ,'linear'))
        for k = 1: (ydimi + ydimj)
          if(k <= ydimi)
            Y(k,:) = classdata(i,k,:);
          else
            Y(k,:) = -classdata(j,k - ydimi,:);
          end
       end
         w0 = [ones(ydimi,1); -ones(ydimj,1)];
         YY = [w0 Y];
         aa= (YY'*YY);
```

```
    w(i,j,:) = (inv(YY'*YY)*YY')*b;

elseif (strcmp(type ,'quadratic'))
 % insert quadratic part
 for k = 1: (ydimi + ydimj)
  if(k <= ydimi)
   % construct quadratic part xixj
   L = length(classdata(i,k,:));
   qindex = 1;
   for findex = 1:L
    for sindex = findex:findex
      quadraticpart(qindex)=classdata(i,k,findex)*classdata(i,k,sindex);
      qindex = qindex + 1;
    end
   end
   Y(k,:) = zeros(1,L+length(quadraticpart));
   Y(k,1:L) = classdata(i,k,:);
   Y(k,L+1:L+length(quadraticpart)) = quadraticpart;
  else
   % construct quadratic part xixj
   L = length(classdata(j,k - ydimi,:));
   qindex = 1;
   for findex = 1:L
     for sindex = findex:findex
       quadraticpart(qindex) =
        classdata(j,k-ydimi,findex)*classdata(j,k-ydimi,sindex);
       qindex = qindex + 1;
     end
   end
   Y(k,:) = zeros(1,L+length(quadraticpart));
   Y(k,1:L) = -classdata(j,k - ydimi,:);
   Y(k,L+1:L+length(quadraticpart)) = -quadraticpart;
  end
```

```
      end
      w0 = [ones(ydimi,1); -ones(ydimj,1)];
      YY = [w0 Y];
      w(i,j,:) = (inv(YY'*YY)*YY')*b;
    else
     error('unknown type');
    end
    end
  end
end


%////////////////////////////////////////////////////////////
% Author: Önder Nazım Onak
%////////////////////////////////////////////////////////////
function [class] = msedclassify(sample,type,weightv)
     if(strcmp(type ,'quadratic'))
         L = length(sample);
         index = 1;
         for i = 1:L
             for j= i:i
                 quadraticpart(index) = sample(i)*sample(j);
                 index = index +1;
             end
         end
         quadraticsample = zeros(1,L + length(quadraticpart));
         quadraticsample(1:L) = sample;
         quadraticsample(L+1:L + length(quadraticpart)) = quadraticpart;
         sample =  quadraticsample;
     end
    [m n] = size(sample);
    % find weight vectors that decribe the separating planes
    [xw yw zw] = size(weightv);
    for i = 1:xw
```

```
        for j = (i+1):yw
            wij(:,1) = weightv(i,j,:);
            distance(i,j)= wij'*[1 sample]';
        end
    end
    output = zeros(xw,1);
    for i = 1:xw
        indices = find(distance(i,:,1) < 0);
        if(isempty(indices))
            output(i) = 1;
        end
    end
    [value class] =  max(output);
    if(value == 0)
        class = xw + 1;
    end
end
```