HANDWRITTEN GURMUKHI CHARACTER RECOGNITION

A Thesis

Submitted In Partial Fulfillment of the Requirements for the Award of the Degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

Kartar Singh Siddharth
(Roll No. 09203008)

Under the Supervision of

Dr Renu Dhir
Associate Professor

and

Mrs Rajneesh Rani
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DR. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY
JALANDHAR-144011 (PUNJAB), INDIA

July, 2011

CERTIFICATE

I hereby certify that the work presented in this thesis entitled "**Handwritten Gurmukhi Character Recognition**" in partial fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science and Engineering submitted in the Department of Computer Science and Engineering, Dr B R Ambedkar National Institute of Technology, Jalandhar, is an authentic record of my own work carried out under the supervision of Dr Renu Dhir and Mrs Rajneesh Rani and refers other researcher"s works which are duly listed in reference section.

The matter presented in this thesis work has not been submitted for the award of any other degree of this or any other university.

[illegible]
Kartar Singh Siddharth
(Roll No. 09203008)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge and belief.

[illegible]
Dr Renu Dhir (Supervisor)
Associate Professor

[illegible]
Mrs Rajneesh Rani (Supervisor)
Assistant Professor

The M.Tech. (thesis) Viva-voice examination of Kartar Singh Siddharth, Roll No. 09203008, has been held on 13.10.2011 and accepted.

[illegible] [illegible]
Supervisor(s)

[illegible]
External Examiner

[illegible]
Head of the Department
Department of Computer Science and Engineering
Dr B R Ambdkar National Institute of Technology, Jalandhar

ACKNOWLEDGEMENT

The successful realization of this thesis work was only possible due to collaboration of many people in many ways. To all of them I would like to pay my gratitude.

First of all, I would like to thank my worthy supervisors duo Dr Renu Dhir, associate professor and Mrs Rajneesh Rani, assistant professor in the department of computer science and engineering at Dr B R Ambedkar National Institute of Technology, Jalandhar. They provided me all possible guidance and support. They suggested me many ideas and solved my puzzles when I was in need.

I would like to thank many students of B.Tech. and M.Tech. particularly, belonged to Punjab region, who cooperated me in collection of handwritten samples of Gurmukhi characters, which I have used in the thesis work. I would like to thank Mr Rakesh Kumar (Care-taker Hostel No. 7), Mr Mohan Singh(Security Guard), and*Fauji Sir* (Mr Ranjeet Singh) for their contribution to write these samples.

I again, would like to thank *Fauji Sir*, lab incharge of our M.Tech. computer labs for well maintenance of lab and computers. I also thank lab incharges of computer center in which I was able to scan my collected samples of Gurmukhi characters.

I would like to thank my M.Tech. classmates who encouraged me throughout the thesis work, specially, Mahesh Jangid for his intimate and informal support and discussion on many related things that we shared.

Finally, I dedicate the thesis to my parents, Sh. Ramesh Chand and Smt. Veervati, for their continuous and endless love, affection, encouragement and support that make everything possible for me to do whatever I am able to. I also thank my younger brother Naresh Chand for his love.

Date: 15th July 2011 (Ashadha Purnima)

Place: Dr B R Ambedkar National Institute of Technology, Jalandhar

Kartar Singh Siddharth

ABSTRACT

Recently there is growing trend among worldwide researchers to recognize handwritten characters of many languages and scripts. Much of research work is done in English, Chinese and Japanese like languages. However, on Indian scripts, the research work is comparatively lagging; most of the research work available is mainly on Devnagari and Bangla scripts. The work on other Indian scripts is in beginning stage.

In this thesis work we have proposed offline recognition of isolated handwritten characters of Gurmukhi script. We have also extended the work by applying the same methodology to recognize handwritten Gurmukhi numerals. In our work we have considered 35 basic characters of Gurmukhi script all assumed to be isolated and bearing header lines on top to recognize. In numerals, handwritten samples of 10 digits from different writers are considered.

The collection of handwritten samples of characters comprises of 200 samples of each of 35 characters, forming total size of 7000 samples. 10 samples of each character are contributed by each of 20 writers of different profiles and educational backgrounds. We have taken all these samples on white papers written in an isolated manner. The dataset used to recognize numerals is collected from 15 different writers each contributing 10 samples of each digit.

After scanning, in preprocessing stage, the samples are converted to gray scale images. Then gray scale image is converted into binary image. We also applied median filtration, dilation, removal of noise having less than 30 pixels, some morphological operations to bridge unconnected pixels, to remove isolated pixels, to smooth pixel boundary, and to remove spur pixels. We segmented these samples in isolated and clipped images of each character based on white spaced pixels used for separation.

We have used 32×32 pixel sized normalized images of characters to extract features. Basically three types of statistical features are considered- zonal density, projection histograms (horizontal, vertical and diagonal) and distance profiles (from left, right, top and bottom side). In addition, directional features of special type named as "Background Directional Distribution (BDD) features" are also used in different combination with the three basic types of features.

The numbers of features extracted and used from these four types of features, namely zonal density, projection histograms, distance profiles and BDD are 16, 190, 128 and 128 respectively. Particularly, in zonal density, we made 16 zones of 4×4 of character image and used density of each zone as feature. In projection histogram we have used horizontal, vertical and both diagonal projection histograms making 190 features in total. In distance profiles we measured distance to first pixels of foreground from four sides- left, right, top and bottom in each row or column. By distance profiles we created 128 features.

In the form of BDD features, first we computed 8 directional features for each of foreground pixel. These features are measured by distribution of background pixels around the foreground pixel in 8 connected directions. To evaluate the weight of background distribution in particular direction specific masks one for each direction are used. For a particular direction, the weight values mapping the weights of background pixels of 8 connected positions in the particular direction are added up. These weight values are specified in the mask corresponding to the particular direction. At second stage, for each zone all feature values in similar directions are summed up to form a single feature in each of 8 directions for a zone. Thus we have obtained 128 BDD features having 8 directional features in each of 16 zones for each character image.

We have derived 10 different sets of feature vectors (or feature sets) by different possible combinations of the four types of features extracted, each used for recognition of samples.

For classification purpose we have used three types of classifiers- Support Vector Machines (SVM), k- Nearest Neighbour (k-NN) and Probabilistic Neural Network to recognize Gurmukhi characters and numerals.

We have obtained all the recognition results with different classifiers and feature sets. The Background Directional Distribution features are the most efficient features and the efficiency obtained with BDD features is enhanced by combining with zoning density, profiling and histogram features. The highest recognition result observed is 95.07% accuracy with combination of feature vector comprising zonal density and BDD features having 144 features and SVM classifier with RBF (Radial Basis Function) kernel. The other combinations of features are resulted either in the form of decreased recognition rate or higher computation time. In numeral recognition we have obtained 99.2% highest recognition rate with feature set comprising of projection histograms and 99.13% rate is obtained with feature set comprising of zonal density and BDD features both with SVM classifier. The results with PNN and K-NN are less significant and in decreasing order of accuracy observed in both character and numeral recognition.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

Chapter 1. INTRODUCTION

The storage of scanned documents have to be bulky in size and many processing applications as searching for a content, editing, maintenance are either hard or impossible. Such documents require human beings to process them manually, for example, postman‟s manual processing for recognition and sorting of postal addresses and zip code. Optical character recognition, usually abbreviated as OCR, translates such scanned images of printed, typewritten or handwritten documents into machine encoded text. This translated machine encoded text can be easily edited, searched and can be processed in many other ways according to requirements. It also requires tinny size for storage in comparison to scanned documents. Optical character recognition helps humans ease and reduce their jobs of manually handling and processing of documents. Computerized processing to recognize individual character is required to convert scanned document into machine encoded form.

In comparison to languages like English and Japanese, the recognition research on Indian languages and scripts is relatively lagging behind. Among available research work on Indian languages, most of the work is on Devnagari and Bangla script. The work on other Indian languages is in fewer amounts.

1.1 CLASSIFICATION OF CHARACTER RECOGNITION SYSTEM

Earlier OCR was widely used to recognize printed or typewritten documents. But recently, there is an increasing trend to recognize handwritten documents. The recognition of handwritten documents is more complicated in comparison to recognition of printed documents. It is because handwritten documents contains unconstrained variations of written styles by different writers even different writing styles of same writer on different times and moods. Sometimes, even a writer can‟t recognize his/her own handwriting, so it is very difficult to gain acceptable recognition accuracy involving all possible variations of handwritten samples.

Traditionally OCR is considered to recognize scanned documents in offline mode. Recently, due to increased use of handheld devices online handwritten recognition attracted attention of worldwide researchers. This online handwritten recognition aims to provide natural interface to users to type on screen by handwriting on a pad instead of by typing using keyboard.

Online recognition recognizes instantly or within a fraction of time just after writing down the sample usually on an electronic pad.

The figure 1.1 depicts the general classification of character recognition system.

Generally all printed or type-written characters are classified in offline mode. The online mode of recognition is mostly used to recognize only handwritten characters.

Off-line handwriting recognition refers to the process of recognizing characters in a document that have been scanned from a surface (such as a sheet of paper) and are stored digitally in grey scale format. After being stored, it is conventional to perform further processing to allow superior recognition. In case of online handwritten character recognition, the handwriting is captured and stored in digital form via different means. Usually, a special pen is used in conjunction with an electronic surface. As the pen moves across the surface, the two- dimensional coordinates of successive points are represented as a function of time and are stored in order.

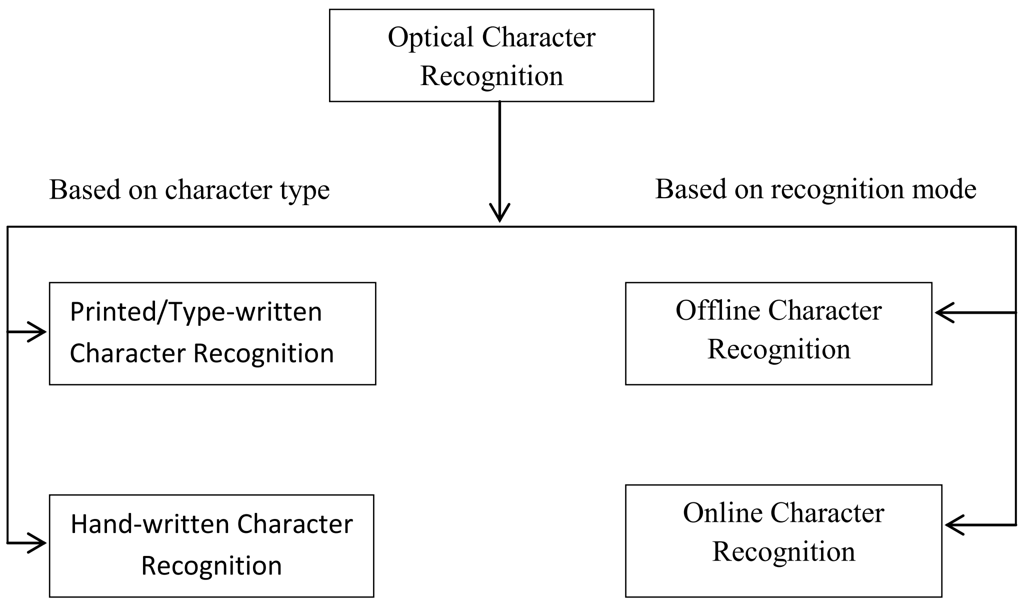

Optical Character Recognition
Based on character type
Based on recognition mode
Printed/Type-written Character Recognition
Hand-written Character Recognition
Offline Character Recognition
Online Character Recognition

Figure 1.1 Classification of character recognition systems

The basic difference between offline and online recognition is that online character recognition has real time contextual and stroke information while offline recognition has only pre-stored static information in the form of scanned image. Online handwritten data is digitized by writing with a special pen device on electronic surface such as digitizer or electronic notepad attached with screen.

1.2 CHARACTER RECOGNITION ARCHITECTURE

Optical character recognition involves many steps to completely recognize and produce machine encoded text. These phases are termed as: Pre-processing, Segmentation, Feature extraction, Classification and Post processing. The architecture of these phases is shown in figure 1.2 and these phases are listed below with brief description. These phases, except post processing are elaborated in next section of overview of OCR phases.

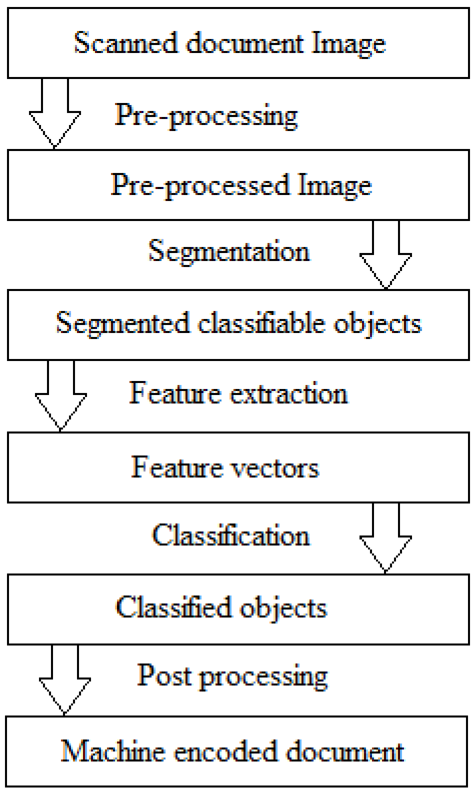


Figure 1.2 Architecture of character recognition system

Pre-processing: The pre-processing phase normally includes many techniques applied for binarization, noise removal, skew detection, slant correction, normalization, contour making and skeletonization like processes to make character image easy to extract relevant features and efficient recognition.

Segmentation: Segmentation phage, which sometimes considered within pre-processing phase itself, involves the splitting the image document into classifiable module object generally isolated characters or modifiers. Generally practiced segmentations are line segmentation, word segmentation, character segmentation and horizontal segmentation to separate upper and lower modifiers particularly in context to most Indian scripts.

Feature Extraction: Feature extraction is used to extract relevant features for recognition of characters based on these features. First features are computed and extracted and then most relevant features are selected to construct feature vector which is used eventually for recognition. The computation of features is based on structural, statistical, directional, moment, transformation like approaches.

Classification: Each pattern having feature vector is classified in predefined classes using classifiers. Classifiers are first trained by a training set of pattern samples to prepare a model which is later used to recognize the test samples. The training data should consist of wide varieties of samples to recognize all possible samples during testing. Some examples of generally practiced classifiers are- Support Vector Machine (SVM), K- Nearest Neighbour (K-NN), Artificial Neural Network (ANN) and Probabilistic Neural Network (PNN).

Post Processing: In post processing step we bind up our work to create complete machine encoded document through the process of recognition, assigning Unicode values to characters and placing them in appropriate context to make characters, words, sentences, paragraphs and finally whole document. We also correct misclassified character based on some linguistic knowledge. We can use dictionary and other language grammatical tools to match the classification results and correct the syntax or semantic of word or sentence. By using dictionary we can restrict the possible combination of characters to form a word, or the combination of words to form a sentence. The recognition and segmentation error can be corrected using lexical information using a lexicon.

1.3 OVERVIEW OF OCR PHASES

1.3.1 Pre-processing

In pre-processing scanned document is converted to binary image and various other techniques to remove noise, to make it ready and appropriate before feature extraction and further computations for recognition are applied. These techniques include segmentation to isolated individual characters, skeletonization, contour making, normalization, filtration etc. Which types of pre-processing techniques will suite; it highly depends on our requirements and also is influenced by mechanism adopted in later steps. However many other basic techniques are commonly applied to all applications, for example, noise removal, because that makes image ease to process and recognize in further steps. Some of the pre-processing techniques are listed and discussed briefly in following sections.

- **Gray Scale Image**

If the input document is scanned in colored image format, It may be required to first convert it into gray scale, before converting to binary image. Gray scale image is in which each pixel shows intensity information that is used to decide how much dark or white the pixel is to be shown. Gray scale image is also called black and white image. These images are composed of shades of gray level varying from black at weakest intensity and white at strongest intensity. This intensity is measured in range of 0 to 1. Gray scale image is monochromatic but multilevel. Multilevel in the sense having multiple shades of gray level for each pixel between range of 0 to 1. While bi-level images have only two levels black or white and are called binary images.

- **Binarization**

Binarization converts coloured (RGB) or gray scale image into binary image. In case of converting coloured image it first needs to convert it into gray image. To convert a gray image into binary image we require to specify threshold value for gray level, dividing or mapping range of gray level into two levels either black or white i.e. either 0 or 1 not between it. In binary image each pixel has one of possible two values 0 and 1, representing black or white respectively. There are many techniques to find threshold value. The popularly used method for threshold value is Otsu"s method [1].

Sezgin and Sankur [2] categorize the threshold methods into the following six groups based on the information the algorithm manipulates: histogram shape, clustering, entropy, object attribute, spatial and local approaches based methods.

- **Smoothing and Noise Removal**

Smoothing operations are used to blur the image and reduce the noise. Blurring is used in pre-processing steps such as removal of small details from an image. In binary images, smoothing operations are used to reduce the noise or to straighten the edges of the characters, for example, to fill the small gaps or to remove the small bumps in the edges (contours)of the characters. Smoothing and noise removal can be done by filtering. Filtering is a neighbourhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighbourhood of the corresponding input pixel. There are two types of filtering approaches: linear and nonlinear, based on the value of an output pixel as a linear and non-linear combination of the values of the pixels in the input pixel"s neighbourhood [3].

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array} \qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Figure 1.3 Two 3 by 3 smoothing (averaging) filter masks. The sum of all the weights (coefficients) in each mask is equal to one

Fore examples, two examples of averaging mask filters (linear approach) are shown in figure 1.3. Each of these filters can remove the small pieces of the noise (salt and pepper noise) in the gray level images; also they can blur the images in order to remove the unwanted details. Normally, these averaging filter masks must be applied to the image a predefined number of times, otherwise all the important features (such as details andedges, etc.) in the image will be removed completely.

- **Skew Detection and Correction**

Deviation of the baseline of the text from horizontal direction is called skew. Document skew often occurs during document scanning or copying. This effectvisually appears as a slope of the text lines with respect to the x-axis, and it mainlyconcerns the orientation of the text lines.In anattempt to partially automate the document processing systems as well as to improvethe text recognition process, document skewangle detection and correction algorithmscan be used. In skew detection skew angle of character string is detected. Some methods rely on detecting connected components and finding the average anglesconnecting their centroids, others use projection profile analysis.[3]

- **Slant Correction**

The character inclination that is normally found in cursive writing is called slant.Slant correctionis an important step in the pre-processing stage of both handwritten wordsand numeral strings recognition. The general purpose of slant correction is to reducethe variation of the script and specifically to improve the quality of the segmentationcandidates of thewords or numerals in a string, which in turn can yield higher recognitionaccuracy.To correct the slant presented first we need to estimate the slant angle (θ), then horizontal shear transform is applied to all the pixels of images of the character/digit in order toshift them to the left or to the right (depending on the sign of the θ).[3]

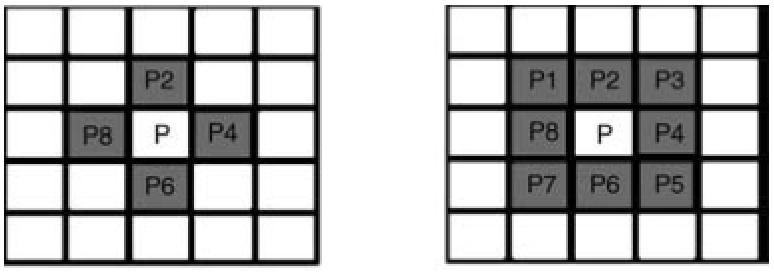
- **Character Normalization**

Character normalization is considered to be the most important pre-processing operationfor character recognition. Normally, the character image is mapped onto a standardplane (with predefined size) so as to give a representation of fixed dimensionalityfor classification. The goal for character normalization is to reduce the within-classvariation of the shapes of the characters/digits in order to facilitate feature extractionprocess and also improve their classification accuracy. Basically, there are two differentapproaches for character normalization: linear methods and nonlinear methods. These methods are described in [3].

- **Contour Tracing/Analysis**

Contour tracing is also known as border following or boundary following. Contourtracing is a technique that is applied to digital images in order to extract the boundaryof an object or a pattern such as a character. The boundary of a given pattern P is defined as the set of border pixels of P. In digital images, each pixel of the image isconsidered as a square. As a result, there are two kinds of border (boundary) pixelsfor each pattern: 4-border pixels and 8-border pixels. Before describing these twotypes of borders in patterns, we define two types of connectivity or neighbourhood indigital images: 4-connectivity and 8-connectivity.

The figure 1.4(a) and 1.4(b) shows pixel P surrounded by neighbouring pixels in 4-connectivity and 8-connectivity approaches respectively.



(a) (b)

Figure 1.4: (a) 4-connected pixel P, and **(b)** 8-connected pixel P

- **Thinning (Skeleton)**

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output. The skeleton obtained musthave the following properties: must be as thin as possible, connected and centered. When these properties are satisfied, the algorithm must stop. Asimple example of thinning of a simple binary image is shown in figure 1.5.

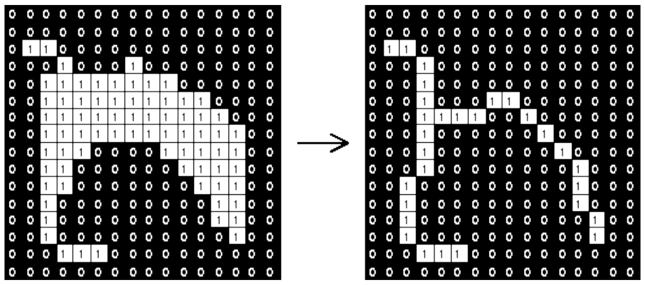


Figure 1.5 Skeletonization by morphological thinning of a simple binary shape. resulting skeleton is connected

Two examples of thinning examples are Hilditch‟s algorithm and Zhang-Suen thinning algorithm [3]. These algorithms are easy to implement.

1.3.2 Segmentation

Segmentation partitions the digital image into multiple segments. These segments form the local zones of the image in such a way that is useful to extract features for character recognition. It is a process of assigning a label to each pixel in an image such that pixels with same label share certain visual characteristics.

In character recognition generally we need following types of segmentation processes one after another sequentially proceeding to segmentation into smaller objects. These segmentation processes are discussed below. The first three types of segmentation are discussed in [4]

Line Segmentation: However straight text lines having enough thoroughly horizontal white space between lines, can be segmented easily, but in practice, particularly in the case of handwritten texts, this technique cannot succeed. I such a skewed, touching or overlapped text we need to discover and apply new techniques. In context of Indian script having header lines a prevalent approach is to detect lines by horizontal projections. Header line is used to have maximum number of pixels while base line is used to have minimum number of pixels in horizontal projections.

Word Segmentation: After line segmentation, the text in each line is segmented to detect words. It is called word segmentation. Word segmentation is easier than line segmentation, because there is generally enough space presents between words and each word is bounded by header line having no space within word.

Character Segmentation: Now after word segmentation, each word needs to segment into characters. It is called character segmentation. In Indian script having header lines we generally, in most practices, use to remove header line first to have vertical space between characters within the word under consideration. Then we segment the word into characters based on present vertical space between these characters.

Zone Segmentation: In most Indian scripts like Devnagari, Gurmukhi, Bangla there are modifiers presented either above the header line or below the base line. The basic and conjunct characters are presented in the middle horizontal zone of below the header line and above the base line. Most Indian researchers prefers to segment the text horizontally in three zones namely upper, middle and lower zones. Thus we further need to segment each character in three horizontal zones. Header line and base line, having maximum and minimum number of pixels in horizontal projections respectively are used to separate these zones.

A typical example of these three zones is shown in figure 1.6 in the context of Gurmukhi script.

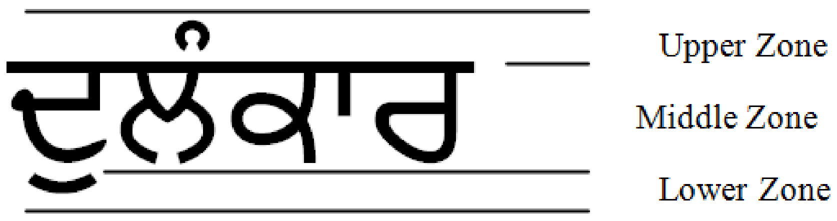


Figure 1.6 Three horizontal zones in Gurmukhi script

By such segmentation at modifier level we have to consider different classes consisting of all the basic characters and lower and upper modifiers. We can extend the work upto half and conjunct characters.

In case of touching, overlapping, faded or broken characters there is need of more complex and advanced algorithms for segmentation.

Finally by segmentation we produce the objects which need to extract certain features and then to classify in pre specified classes based on these features using different classifiers.

1.3.3 Feature Extraction

Feature extraction is extracting information from raw data which is most relevant forclassification purpose and that minimizes the variations within a class and maximizes thevariations between classes.

Selection of a feature extraction method is probably the single most important factor inachieving high recognition performance in character recognition systems. Different featureextraction methods are designed for different representations of the characters, such as solidbinary characters, character contours, skeletons (thinned characters) or gray-level subimagesof each individual character. The feature extraction methods are discussed in terms ofinvariance properties, reconstructability and expected distortions and variability of thecharacters.

In feature extraction stage each character is represented by a feature vector, whichbecomes its identity. The major goal of feature extraction is to extract a set of features, whichmaximizes the recognition rate with the least amount of confusion. Different feature extraction methods fulfil these criteria at varying degree and a particularfeature extraction method cannot be useful similarly in different domains.

Feature Extraction Methods

Some of the mostly and generally practiced feature extraction methods based on statistical, structural and manymore features, properties and characteristics of images are discussed in following sections in brief.

- ***Zoning:***

In zoning, the character image is divided into $N \times M$ zones. From each zone features are extracted to form the feature vector. The goal of zoning is to obtain the local characteristics instead of global characteristics[8]. From each zone obtained after zoning many local features can be derived, for example, density and directional features.

Density is considered as the ratio of number of foreground pixels to the number of total pixels in a particular zone. In addition to density features many other features like directional features as directional histogram using skeleton or contour can be used.

- ***Projection Histograms***

The basic idea behind using projections is that character images, which are 2-D signals, can be represented as 1-D signal. These features, although independent to noise and deformation, depend on rotation. Projection histograms count the number of foreground pixels in each column and row of a character image. Exceptional to typical horizontal and vertical histograms, diagonal and radial histograms also can be used [8].

- ***Profiles***

The profile counts the number of pixels (distance) between the bounding box of the character image and the edge of the character. The profiles describe well the external shapes of characters and allow distinguishing between a great number of letters, such as “ਖ”, “ਬ”,“ਧ”and “ਪ”.Profiles also can be applied to contour or skeleton of images. In some situations in addition to outside profiles we can be computed inside profiles also. [8]

- ***Crossings and Distances***

Crossings count the number of transitions from background to foreground pixels along vertical and horizontal lines through the character image and Distances calculate the distances of the first image pixel detected from the upper and lower boundaries, of the image, along vertical lines and from the left and right boundaries along horizontal lines. We can apply these crossing and distance features either on all or some selected rows and columns distributed on entire sample. [8]

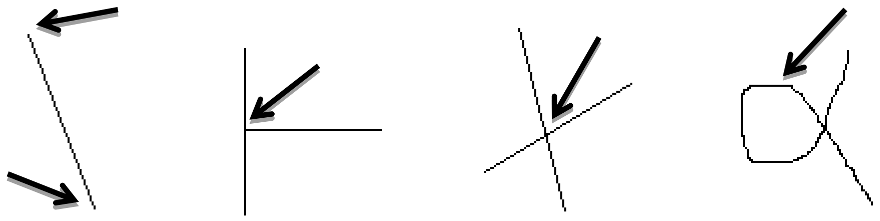
- ***Structural or Topological Features***

Structural features are based on topological and geometrical properties of the character, such as aspect ratio, cross points, loops, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom and curvature information. (See figure 4.5) [3], [8]

In a thinned image, black pixels that have a number ofblack neighbors not equal to 0 or 2 are given the name of **feature points**. Feature pointshave been classified into three types:

- End points
- Branch points, and
- Cross points

The first type of point, the end point, is defined as the start or the end of a line segment.A branch point is a junction point connecting three branches. A cross point is anotherjunction point that joins four branches. Figure 1.7 visually illustrates the threetypes of feature points and loop.



(a) End points **(b)** Branch point**(c)** Cross point**(d)** Loop

Figure 1.7 Feature points and loop (indicated by arrow)

Curvature of a given curve or binary contouris another essential local feature. In calculus, the curvature c at a point$p$on acontinuous plane curve C is defined as $c = lim_{\Delta s \to 0} \frac{\Delta\alpha}{\Delta s}$,where s is the distance tothe point p along the curve and $\Delta\alpha$ is the change in the angles of the tangents to thecurve at the distances s and $s + \Delta s$, respectively. [3]

Sometimes Projection histogram and profile features are also considered as structural features.

- ***Moments***

Moments and functions of moments have been employed as pattern features in numerous applications to recognize two-dimensional image patterns. These pattern features extract global properties of the image such as the shape area, the center of the mass, the moment of inertia, and so on. The general definition of moment functions m_{pq} of order $(p + q)$for a $X \times Y$ digital image intensity function $f\ (x, y)$ is as follows:

$$m_{pq} = \sum_y \sum_x \psi_{pq}(x, y) f(x, y),$$

where p, q are integers between (0,∞) and represent the order, x and y are the x- andy-pixel of the digital image, and $\psi_{pq}(x, y)$ is the basis function.[3]

A desirable property for any pattern feature is that it must preserve informationunder image translation, scaling, and rotation.Hu was the first to develop a number ofnonlinear functions based on **geometric moments** thatwere invariant under image transformation.

Zernike defined a completeorthogonal set of complex polynomials over the polar coordinate spaceinside a unit circle (i.e., $x^2 + y^2 = 1$). For digital image, **Zernike moments** are the projection of the image intensity function $f\ (x, y)$ onto the complex conjugate of Zernike polynomial. Further, Bhatia and Wolf derived **pseudo-Zernike moments** from Zernike moments.

The examples of other moments used are **Legendre moment** and **Tchebichef moments**. A detailed description for these moments can be found in [3].

- ***Directional Features***

Characters comprise strokes that are orientedlines, curves, or polylines. The orientationor direction of strokes plays an important role in differentiating between variouscharacters.For statistical classification basedon feature vector representation, charactershave also been represented as vectorsof orientation/direction statistics. To do this, the stroke orientation/direction angle ispartitioned into a fixed number of ranges, and the number of stroke segments in eachangle range is taken as a feature value.

Both orientation and direction histogram features can be called direction featuresin general.The local stroke orientation/direction of a character can be determined in different ways: skeleton orientation, stroke segment, contour chaincode, gradient direction, and so on. The contour chaincode and gradient directionfeatures are now widely adopted because they have simple implementation and areapproximately invariant to stroke-width variation. The decomposition of directional features using contour and gradient is briefly described below. More details about direction features can be found in [3].

Contour Chaincode Decomposition: The contour pixels of a binary image are commonly encoded into chaincodes whenthey are traced in a certain order, say counterclockwise for outer loop and clockwisefor inner loop. Each contour pixel points to its successor in one of eight directions as shown in Figure 1.8. As the contour length is approximately invariant to stroke width,and the chaincode direction reflects the local stroke direction, it is natural to assigncontour pixels of the same chaincode to a direction plane and extract feature valuesfrom the direction planes.[3]

There are basically two ways to determine the direction codes of contour pixels.In one way, the order of contour pixels is obtained by contour tracing, and then the chaincodes are calculated. Contour tracing, however, is not trivial to implement.Another way examines

a 3 $\times$ 3 window centered at each contour pixel (a black pixelwith at least one of the 4-neighbors being white). The contour pixel is assignedto one or two directions according to the configuration of neighbourhood.

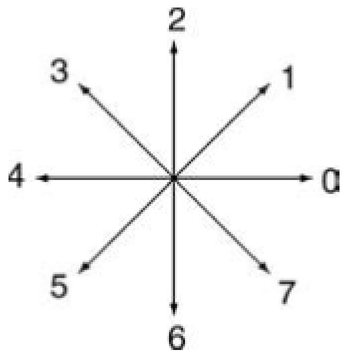


Figure1.8 Eight chaincode directions

For extracting orientation feature, the eight direction planes are merged into fourorientation planes by$f_i(x,y) = f_i(x,y) + f_{i+4}(x,y), i = 0,1,2,3.$

Gradient Direction Decomposition: Considering that the stroke edge direction is approximately normal to the gradientof image intensity, we can alternatively decompose the local gradient into directionalplanes and extract the gradient direction feature. Unlike the chaincode feature, whichis applicable to binary images only, the gradient feature is applicable to gray-scaleimages as well and is more robust against image noise and edge direction fluctuations.The gradient can be computed by the Sobel operator, which has two masks for thegradient components in horizontal and vertical directions, as shown in Figure 1.9.

+1	+2	+1
0	0	0
-1	-2	-1

-1	0	+1
-2	0	+2
-1	0	+1

Figure1.9 Sobel masks for x- and y- gradient

By Sobel mask horizontal and vertical gradients (g_x *and* g_y) at pixel(x,y), which are horizontal and vertical derivative approximations, are computed by following:

$$g_x(x,y) = f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1) - f(x-1,y-1) \\ - 2f(x-1,y) - f(x-1,y+1),$$

$$g_y(x,y) = f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1) - f(x-1,y-1) - 2f(x,y-1) - f(x+1,y-1)\,.$$

The resulting gradient approximations can be combined to give the gradient magnitude, using $|g(x,y)| = \sqrt{(g_x(x,y))^2 + (g_y(x,y))^2}$ and we can calculate the gradient direction by angle $\theta = tan^{-1}(g_y(x,y)/g_x(x,y)$. The gradient angle having arbitrary direction can be transformed into specified standard directions by decomposing gradient vector into two components coinciding with the two neighbouring standard directions. (Fig. 1.10)[3]

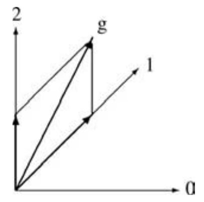


Figure1.10 Decomposition of gradient vector to map into standard direction (either 1 or 2 in this case)

- ***Image Registration***

In image processing applications, image registration is a necessity as it allows us tocompare and to integrate pictures taken at different times, from different sensors or from different viewpoints. Combining these images provides a comprehensiveand global picture.

Image registration is defined as a mapping between two ormore images, both spatially and with respect to intensity. Consider two 2-dimensionalimages (2D) with intensity functions, $f_1(x,y)$ and $f_2(x,y)$. Both ofthese functions map to their corresponding intensity values in their respective images.Therefore, the mapping between two images can be described as follows:

$$f_2(x,y) = g\left(f_1\big(T(x,y)\big)\right),$$

where $f_1(x,y)$ and $f_2(x,y)$ are the intensity functions of two 2D images, T is a 2D spatial-coordinate transformation and g is a one dimensional intensity transformation.

The objective of image registration is to determine the optimal spatial and intensitytransformation, that is, T and g, respectively, in order to match images for (1)determining the parameters of the matching transformation or (2) exposing importantdifferences between images. [3]

In [3] four classes ofregistration methods: correlation, Fourier methods, point mapping, and elastic methods are described.

- ***Hough Transform***

A popular feature extraction method used in digital image processing is the Houghtransform (HT). It is able to detect straight lines, curves, or any particular shape thatcan be defined by parametric equations. In essence, this method maps the figure pointsfrom the picture space to the parameter space and, thereafter, extracts the features.There are two types of Hough transform: standard HT and randomized HT.[3]

In the standard Hough transform, all figure pointsfrom the picture space are mapped to the parameter space.Randomized Hough transform is anenhanced approach for detecting features in binary images. It takes into account thedrawbacks of standard Hough transform, namely long computation time and largememory requirements.[3]

- ***Line Based Representation***

Usually the strokes in characters are thicker than single pixel. We can represent this thicker representation in the form of contour (edges) or skeleton. Such formed strokes are single pixel wide and this representation is called line based representation.

Edge (Contour): Anedge is defined as the boundary between two regionswith different gray-level properties. It is possible to detect an edge because thegray-level properties of an image at an edge drastically change, which results ingray-level discontinuity.

In most edge-detection methods, the local first and second derivatives are computed.A drastic change in the magnitude of the first derivative signifies the presenceof an edge in an image. Similarly, using the sign of the second derivative, it is possibleto determine whether an edge pixel lies on the dark or on the light side of an edge. Thesecond derivative is positive for a dark side and negative for a light side. Therefore,if a second derivative of a given boundary in an image is a negative spike followedby a positive spike, then there is a detection of a light-to-dark edge[3].

Tracing a contour involves the detection of edges. As a contour is composed of aset of boundary points, it is possible to extend any edge-detection method to detectcontours.

Thinning (skeletonization): Thinning is a process during which a generally elongated patternis reduced to a line-like representation. This line-like representation or a collectionof thin arcs and curves is mostly referred to as a skeleton. Most thinning algorithmsare iterative in that they delete successive layers of pixels on the edge of the patternuntil only a skeleton is left. A set of tests is performed to assess if a given blackpixel p is to be deleted or to be preserved. Obviously, the deletion or preservation ofp depends on its neighboring pixels‟‟ configuration. Thinning algorithms are dividedinto two categories: sequential and parallel[3].

In a sequential algorithm, the sequence in which the pixels are tested for deletionis predetermined, and each iteration depends on all the previous iterations. That is,the deletion of p in the nth iteration depends on all the operations that have beenperformed so far, that is, on the result of the *(n − 1)th* iteration as well as on thepixels already processed in the nth iteration.

In a parallel algorithm, the deletion of pixels in the *nth* iterationwould depend onlyon the result that remains after the *(n − 1)th*; therefore, all pixels can be examinedindependently in a parallel manner in each iteration.

- ***Fourier Descriptors***

Fourier descriptor is one of the most popular and efficient methods to retrieve a particular shape from a given image. It represents the shape of the object in the frequency domain. By frequency domain representation Fourier descriptors has many advantage over its counterparts as strong discrimination ability, low noise sensitivity, easy normalization and information preservation [3].

Fourier descriptors are used for describing the shape (closedcurve) of any object found on an input image. However, before computing the Fourierdescriptors, the input image is digitalized and boundary information of the object is extracted and normalized. During normalization, the data points of the shapeboundary of the object and model are sampled to have the same number of datapoints. As Fourier descriptors require a 1D representation of the boundary information,shape signatures are used. A shape signature maps a 2D representation of ashape to a 1D representation.Complexcoordinates, centroid distance, curvature, and cumulative angular function are the examples of shape signatures used in Fourier descriptors [3].

Two types of FD: standard Fourier descriptors and elliptical Fourier descriptors are described in [3].

- ***Shape Approximation***

The contour or skeleton of a character image can be represented as approximation into piecewise line segments, chaincodes and smoth curve. Line segment representation changes the character image into approximated polygon. Polygonal approximation and chaincodes are discussed here.

Polygonal Approximation: In polygonal approximation character image contour or skeleton is segmented into small piecewise straight line segment and all these line segments form a polygon by joining each other. This polygonal representation is an approximation of character image. If we want the polygonal representation with low information loss then the lines segment should be as large as possible and this number can be upto the number of pixel points in character contour or skeleton.

Chaincodes: Chaincodes are used to represent a shape boundary by a connected sequence of straight lines that have fixed length and direction. The direction of line is based on 4- or 8-connectivity (see figure 1.11(a) and 1.11(b)) and coded with the code corresponding to direction of that particular fraction of line.

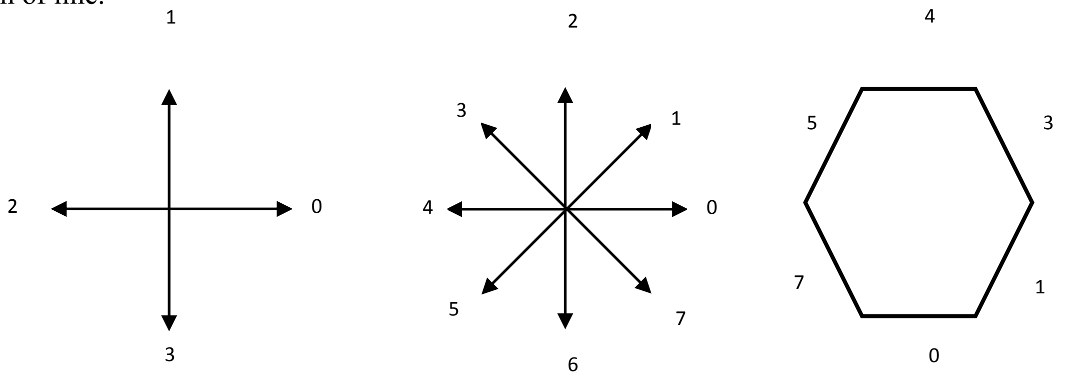


Figure1.11 Chaincodes: **(a)** 4-connectivity, **(b)** 8-connectivity, **(c)** example

For example, the polygon given in figure 1.11(c) can be represented by 8-connectivity chaincode 013457 starting from bottom horizontal line segment. Similarly any shape can be represented by 8-connectivity or 4-connectivity scheme. The shape of character can also be represented by Bezier curve and its generalization B-splines[3].

Feature Extraction Steps

We can decompose the problem of feature extraction in two steps: First feature construction and second is feature selection.

1. Feature Construction

Feature construction is one of the key steps in the data analysis process,largely conditioning the success of anysubsequent statistics or machine learning endeavour. Inparticular, one should beware of not losing information atthe feature construction stage. It may be a good idea to addthe raw features to the pre-processed data or at least tocompare the performances obtained with eitherrepresentation. The medical diagnosis example that wehave used before illustrates this point. Many factors mightinfluence the health status of a patient. To the usualclinical variables (temperature, blood pressure, glucoselevel, weight, height, etc.), one might want to add dietinformation (low fat, low carbonate, etc.), family history,or even weather conditions. Adding all those featuresseems reasonable but it comes at a price: it increases thedimensionality of the patterns and thereby immerses therelevant information into a sea of possibly irrelevant, noisyor redundant features[6].

2. Feature Selection

Feature selection is primarily performed to selectrelevant and informative features; it can have othermotivations, including[6]:

1. general data reduction, to limit storage requirementsand increase algorithm speed.
2. feature set reduction, to save resources in the nextround of data collection or during utilization;
3. performance improvement, to gain in predictive accuracy;
4. data understanding, to gain knowledge about theprocess that generated the data or simply visualize the data

Based on the relationship with classifiers,feature selection approaches can generally be divided into four categories: filter, wrapper, embedded and hybrid [3].

A **filter** approach operates independently of any classification algorithm. Itevaluates feature subsets via their intrinsic properties. These properties act as an approximation of the value ofthe features from the point of view of a “generic classifier”. Irrelevant/redundant featuresare “filtered” out before the features are fed into the classifier.

On the contrary, a**wrapper** approach is normally tied to a classifier, with the feature selection processwrapped around the classifier, whose performance directly reflects the goodness ofthe feature subsets. In a wrapper, the feature selection process is wrapped around a learning algorithm.The feature subsets are evaluated via the performance of the learning algorithm. Awrapper method naturally entails training and implementation of learning algorithmsduring the procedure of feature selection.[3].

The main problem of a wrapper is that its cost may be prohibitively high because itinvolves repeated training and implementation of the classifier. The two techniques used for performance evaluation using training and testing (i.e. machine learning) are: *k*-fold cross validation and hold-out validation. *k*-fold cross-validation partitions available data into k sets.In each run k − 1 sets are used for training and the remaining kth set is used for testing. Finally, an average accuracy over k runs is obtained. The hold-out test, on thecontrary, only involves one run in which 50–80% of total data is used for training andthe remainder for testing. Although more efficient, the hold-out test is not statisticallyoptimal[3].

Filters and **wrappers** differ mostly by the evaluationcriterion. It is usually understood that filters use criterianot involving any learning machine, e.g. involving a relevance indexbased on correlation coefficients or test statistics, whereaswrappers use the performance of a learning machinetrained using a given feature subset. Both filter andwrapper methods can make use of search strategies toexplore the space of all possible feature combinations that is usually too large to explore exhaustively. Yet filtersare sometimes assimilated to feature ranking methods forwhich feature subset generation is trivial since only singlefeatures are evaluated[6].

In the **embedded** approach, feature selection is embedded intothe classifier. Both of them are implemented simultaneously and cannot be separated. In embedded approaches, feature selection is tied tightly to, and is performedsimultaneously with, a specific classifier. When the classifier is constructed,the feature selection, hence dimensionality reduction, is achieved as a side effect.Decision trees are typical examples[3].

Hybrid form of feature selection utilizes aspects of both the filter and wrapper approaches in the hope of enjoying thebenefits of both. In hybrid approach generally filter is followed by wrapper but its reverse can be true is some situations. In first way, filter can be used to generate a ranked list of features. On thebasis of the order thus defined, nested subsets of features can be generated and computed by a learning machine, i.e.following by a wrapper approach[3],[6].

1.3.4 Classification

Feature extraction is used to extract the relevant features which are used in classification stage to map the selected features of a character image to a particular class or label of characters. Thus final goal of recognition is to obtain the class labels of character patterns. In order to obtain such class labels a variety of classification methods or classifiers can be applied.

For integrating the classification results with contextual informationlike linguistics and geometrics, the outcome of classification is desired tobe the membership scores (probabilities, similarity or dissimilarity measurements) ofinput pattern to defined classes rather than a crisp class label.

There are many types of classification methods, including statistical methods (parametric and non-parametric), structural methods(string and graph matching), Artificial Neural Networks (ANNs) (supervised and unsupervised learning), Support Vector Machines (SVMs) and combinationof multiple classifiers. Statistical methods, ANNs and SVMs input a featurevector of fixed dimensionality mapped from the input pattern. These different types of classification methods are briefly described here[3].

- **Statistical Classification Methods**

Statistical classification methods are based on the Bayes decision theory, which aims tominimize the loss of classification with given loss matrix and estimated probabilities.According to the class-conditional probability density estimation approach, statisticalclassification methods are divided into parametric and nonparametric ones.Parametric methods assume a functional form for the density of each class e.g. Gaussian classifiers. Nonparametric classifiers do not assume any functional form for the conditional distributions. The two approaches related to non-parametric classifiers are k-nearest neighbour and Parzen widow.

- **Artificial Neural Networks**

Artificial neural networks have similarity to neuron system. A neural network is composed of anumber of interconnected neurons, and the manner of interconnection differentiatesthe network models into feedforward networks, recurrent networks, self-organizingnetworks, and so on. Single layer neural network and multilayer perceptron are the examples of ANN classifiers.

- **Support Vector Machines (SVM)**

Support Vector Machine is relatively new classification technique producing efficient recognition results. SVM is a new type of hyperplane classifier developed based on statistical learning theory given by Vapnik. Basically SVM is a binary (two class) linear classifierin kernelinducedfeature space and is formulated as a weighted combination of kernel functionson training examples.The kernel function represents the inner product of twovectors in linear/nonlinear feature space. Multiclass classification is accomplished by combiningmultiple binary SVM classifiers.

- **Structural Classification Methods**

Structural methodsrecognize patterns via elastic matching of strings, graphs, or other structural descriptions.Structural pattern recognition methods are used more often in online character recognitionthan in offline character recognition. Instead of representing the character pattern as feature vector of fixed dimensionality, structural methods represent a pattern as a structure (string, tree, or graph) offlexible size.The structural representation records the stroke sequence or topologicalshape of the character pattern, and hence resembles well to the mechanism of humanperception. In recognition, each class is represented as one or more structural templates,the structure of the input pattern is matched with the templates and is classifiedto the class of the template of minimum distance or maximum similarity. Despite that the automatic learning of structural modelsfrom samples is not well solved, structural recognition methods have some advantagesover statistical methods and neural networks: They interpret the structure ofcharacters, store less parameters, and are sometimes more accurate. There are two widely used methods of structural features: attributed string matching andattributed graph matching.

- **Multiple Classifier Methods**

By multiple classifier methods, the classification results of multiple classifiersare combined to reorder the classes. Different classifiers show varying performance:varying classification accuracy and speed, and different errors on concrete patterns. Hence, it is natural to combine the strengths of different classifiers to achieve higherperformance. This is usually accomplished by combining the decisions (outputs) ofmultiple classifiers. In research, many combination methods have beenproposed, and the applications to practical problems have proven the advantage ofensemble over individual classifiers. The combination of different classifiers can be categorized as parallel (horizontal) and sequential (vertical, cascaded). Parallel combination is more often adopted for improving the classification accuracy,whereas sequential combination is mainly used for accelerating the classification oflarge category set.

1.4 ISSUES OF HANDWRITTEN CHARACTER RECOGNITION

To recognise handwritten documents, either online or offline, the character recognition is much affected by style variations of handwriting by different writers and even different styles of same writer on different times. Distortion and noise incorporated while digitization is also a major issue in character recognition that affects the recognition accuracy negatively. The many character recognition issues regarding handwritten character recognition are listed below:

1.4.1 Handwriting Style Variations

Different writers and even same writer have different handwriting styles. Many times a person finds himself/herself unable to recognize his/her own handwriting. Hence, practically it is much difficult to recognize handwriting by machine efficiently. Deformed geometry, slants, skews, overlapping, noise, distortion are inserted by different writers in different ways. Geometric properties like aspect ratio, position and size vary.

One can note that some kind of variations also exists in each sample of a character although such samples share high degree of similarities. The shape of a character is also influenced by the word in which it is appearing. Characters can look similar although their number of strokes, and the drawing order and direction of the strokes may vary considerably.

1.4.2 Constrained and Unconstrained Handwriting

The characters to recognize may be constrained or unconstrained. Because unconstrained documents include all possible style variation, so such document are much difficult to recognize. To make this recognition process easier atleast for laboratory work, constrained documents are practiced to recognize.

In constrained document format, the handwritten samples are written in standard format that make the characters easy to recognize. The constrained document has box discrete and space discrete nature of characters or words. In box discrete nature, each character is written in separate standard sized box. In space discrete nature, characters are written have much space from one-another to make the segmentation and recognition easier.

Unconstrained documents consist of touching, overlapping and cursive characters. Cursive writing makes recognition difficult due to stroke variability. Touching characters are difficult to segment the characters from each other and overlapping characters makes this situation worse.

1.4.3 Writer Dependent or Independent Recognition

Writer dependent recognition system is used to recognize the samples of only those writers whose samples are takes to train the recognition system. That is, written dependent systems are specific to a group of writers. In writer dependent system, all possible style variations can be trained to system, hence a higher recognition rate can be obtained.

At the other hand, writer independent system needs the generalization of the recognition system also to recognize the handwritten samples of unknown writers. Hence, it needs to train the system with all possible and commonly used style variations. Hence it need to train the system with large number of samples taken from large number of different types of writers, to make the recognition system generalized. Due to recognition of unknown samples, the recognition rate of writer independent system is comparatively lower. In practice, writer independent systems are in more demand because of generalized application.

1.4.4 Personal and Situational Aspects

Personal factors include writer"s writing style which might be affected byhandedness- either left handed or right handed and. Many persons are habitual to write random or specific inclined text lines. A good recognition requires neat and clean handwriting and this writing style also depends on profession of writers to some extent.

The situational aspects depend on the facts either writer is interested or not to write, how much attention a writer is paying, text is written giving proper time or in hurry, whether there was any interruption while writing, how was the quality of material (e.g. paper and pencil) used for writing etc.

1.5 APPLICATION OF HANDWRITTEN CHARACTER RECOGNITION

Handwritten character recognition system is basically used to convert a sample image character to corresponding machine coded character. This basic characteristic can be used to derive many other practical applications. Some of these applications are listed below:

- **Postal address and code reading:**The manual distinction and sorting of letters on the basis of specified addresses in a post office takes much time and labour. Offline handwritten character recognition can be used to recognize postal address and pin code etc. on letters. On the basis machine coded address these letters can be sorted and processed easily.
- **Check reading:** The abundant numbers of checks are need to process in banks. Handwritten or printed OCR system can be used for automatically reading the name of recipient, signature verification, filled amount reading and reading all other information.

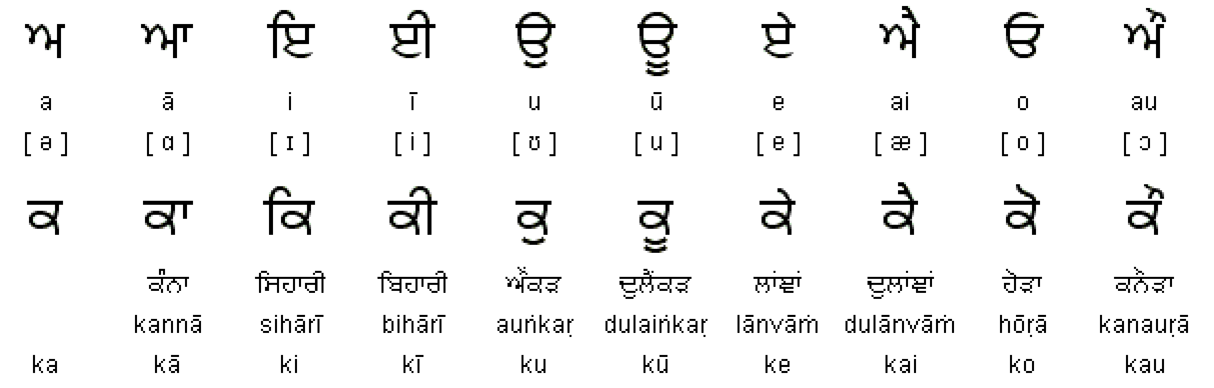
- **Form processing:**Form processing can be used to process forms and documents of public applications. In such forms handwritten information is written in space provided. This handwritten information can be processed by Handwritten OCR system automatically.
- **Signature verification:** On legal or other documents that include the signatures of authorities and persons, signature can be verified using handwritten character recognition system. The system once can be trained by various samples of signatures of required persons and later on any document their signatures can be verified.

1.6 OVERVIEW OF GURMUKHI SCRIPT

The Gurmukhi alphabet was devised during the 16th century by Guru Nanak, the first Sikh guru, and popularized by Guru Angad. The word Gurmukhi means the sayings came out from the mouth of (Sikh) Gurus [9]. Gurmukhi script, also spelled as Gurumukhi, is used to write primarily Punjabi and secondarily Sindhi language. Guru Angad not only modified, rearranged the letters but also documented them properly into a script [10].

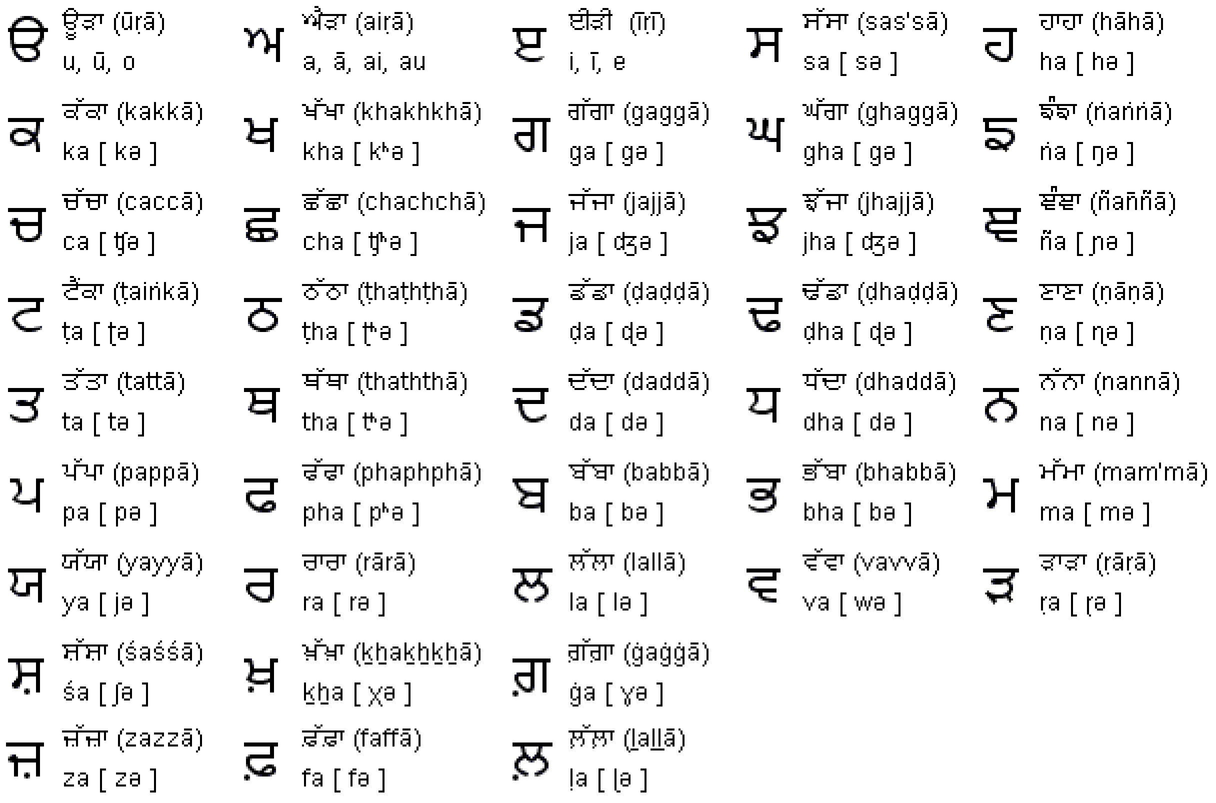
Gurmukhi script consists of 35 basic characters, among which there are 3 vowel holders. In addition to these 35 characters, there are 12 vowel modifiers, 6 additional modified consonants, forming 41 consonants with 35 basic characters [11][12]. In figure 1.12 the Gurmukhi script alphabet is shown.

The 35 characters shown in first 7 rows of figure 1.12(b) are basic characters of Gurmukhi scripts. The characters in rows 8 and 9 were added later. The basic characters are arranged in different rows of 5 characters each representing different sections or *vargas.* Among these basic characters, first three characters of first row are called vowel careers, vowel holders, or *matra vahak.* These are the only characters that carry all vowels on themselves. Modifiers (*matras*) of different vowels are imposed on all other characters known as consonants. These modifiers are joined with a character mostly either in top or bottom horizontal zones, while in middle zone character itself resides. An example of Gurmukhi word depicting three horizontal zones is shown in figure 1.13.



ਅ	ਆ	ਇ	ਈ	ਉ	ਊ	ਏ	ਐ	ਓ	ਔ
a	ā	i	ī	u	ū	e	ai	o	au
[ə]	[ɑ]	[ɪ]	[i]	[ʊ]	[u]	[e]	[æ]	[o]	[ɔ]
ਕ	ਕਾ	ਕਿ	ਕੀ	ਕੁ	ਕੂ	ਕੇ	ਕੈ	ਕੋ	ਕੌ
	ਕੰਨਾ	ਸਿਹਾਰੀ	ਬਿਹਾਰੀ	ਔਂਕੜ	ਦੁਲੈਂਕੜ	ਲਾਂਵਾਂ	ਦੁਲਾਂਵਾਂ	ਹੋੜਾ	ਕਨੌੜਾ
	kannā	sihārī	bihārī	auṅkaṛ	dulaiṅkaṛ	lānvāṁ	dulānvāṁ	hōṛā	kanauṛā
ka	kā	ki	kī	ku	kū	ke	kai	ko	kau

(a) Vowels and Vowel diacritics (*Matras*)



ੳ	ਊੜਾ (ūṛā) u, ū, o	ਅ	ਐੜਾ (aiṛā) a, ā, ai, au	ੲ	ਈੜੀ (īṛī) i, ī, e	ਸ	ਸੱਸਾ (sas'sā) sa [sə]	ਹ	ਹਾਹਾ (hāhā) ha [hə]
ਕ	ਕੱਕਾ (kakkā) ka [kə]	ਖ	ਖੱਖਾ (khakhkhā) kha [kʰə]	ਗ	ਗੱਗਾ (gaggā) ga [gə]	ਘ	ਘੱਗਾ (ghaggā) gha [gə]	ਙ	ਙੰਙਾ (ṅaṅṅā) ṅa [ŋə]
ਚ	ਚੱਚਾ (caccā) ca [tʃə]	ਛ	ਛੱਛਾ (chachchā) cha [tʃʰə]	ਜ	ਜੱਜਾ (jajjā) ja [dʒə]	ਝ	ਝੱਜਾ (jhajjā) jha [dʒə]	ਞ	ਞੰਞਾ (ñaññā) ña [ɲə]
ਟ	ਟੈਂਕਾ (ṭaiṅkā) ṭa [ʈə]	ਠ	ਠੱਠਾ (ṭhaṭhṭhā) ṭha [ʈʰə]	ਡ	ਡੱਡਾ (ḍaḍḍā) ḍa [ɖə]	ਢ	ਢੱਡਾ (ḍhaḍḍā) ḍha [ɖə]	ਣ	ਣਾਣਾ (ṇāṇā) ṇa [ɳə]
ਤ	ਤੱਤਾ (tattā) ta [tə]	ਥ	ਥੱਥਾ (thaththā) tha [tʰə]	ਦ	ਦੱਦਾ (daddā) da [də]	ਧ	ਧੱਦਾ (dhaddā) dha [də]	ਨ	ਨੱਨਾ (nannā) na [nə]
ਪ	ਪੱਪਾ (pappā) pa [pə]	ਫ	ਫੱਫਾ (phaphphā) pha [pʰə]	ਬ	ਬੱਬਾ (babbā) ba [bə]	ਭ	ਭੱਬਾ (bhabbā) bha [bə]	ਮ	ਮੱਮਾ (mam'mā) ma [mə]
ਯ	ਯੱਯਾ (yayyā) ya [jə]	ਰ	ਰਾਰਾ (rārā) ra [rə]	ਲ	ਲੱਲਾ (lallā) la [lə]	ਵ	ਵੱਵਾ (vavvā) va [wə]	ੜ	ੜਾੜਾ (ṛāṛā) ṛa [ɽə]
ਸ਼	ਸ਼ੱਸ਼ਾ (śaśśā) śa [ʃə]	ਖ਼	ਖ਼ੱਖ਼ਾ (k̲h̲ak̲h̲k̲h̲ā) k̲h̲a [χə]	ਗ਼	ਗ਼ੱਗ਼ਾ (ġaġġā) ġa [ɣə]				
ਜ਼	ਜ਼ੱਜ਼ਾ (zazzā) za [zə]	ਫ਼	ਫ਼ੱਫ਼ਾ (faffā) fa [fə]	ਲ਼	ਲ਼ੱਲ਼ਾ (l̲al̲l̲ā) l̲a [ɭə]				

(b) Consonants (*Vianjans*), however first three characters are vowel careers
Figure1.12 Alphabet of Gurmukhi Script (Courtesy [9])

Last two characters in the first row of basic characters are making the row upto 5 and this first row is represented as *mul varg*. The five rows in the middle of basic characters are organised quite cleverly and understanding what is going on here will help you learn them easier. These sections are organized row wise according to pronunciation from different parts of mouth as well as these are arranged column wise distinguishing the stress and way of pronunciation [11], [13]. The sections (*vargas*)of these 5 middle rows are named on the basis of first characters of these rows as ‚*ka varg*‟, ‚*cha varg*‟, ‚*ṭa varg*‟, ‚*ta varg'*and*'pa varg'*respectively.The last row of basic characters is known as ‚*antim varg*‟.

Additionally, Gurmukhi alphabet contains some symbols, which are used as modifier of characters. These symbols are *tippi* (ੰ), *bindi* (ਂ), *adhak* (ੱ), *halant* (੍) and *visarg* (ਃ).

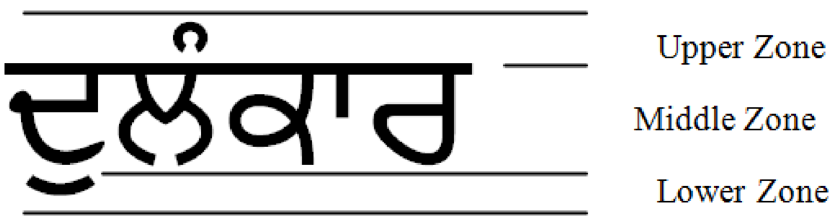


Figure1.13 Three horizontal zones in Gurmukhi script

Characteristics of Gurmukhi Scripts

Gurmukhi script contains following listed characteristics[9][13]:

- Gurmukhi alphabet is syllabic in which all consonants have an inherent vowel. Gurmukhi is written from left to right like most other Indian and Roman scripts.
- Unlike Roman characters, Gurmukhi is written below the line and it has no concept of lower and upper case.
- A text of Gurmukhi script can be partitioned into three horizontal zones namely upper zone, middle zone and lower zone. (Figure 1.13)
- Normally consonants appear in middle zone and diacritics (vowel modifiers, half characters and other modifiers and symbols) appear in upper zone, lower zone, before or after the consonant to change syllable sound of that particular consonant.
- The Gurmukhi script, unlike the Greek and Roman alphabets, is arranged in a logical fashion: vowels first, then consonants (Gutturals, Palatals, Cerebrals, Dentals, and Labials) and semi-vowels.
- Upper zone and middle zone is separated by a line called header line or sirorekha as present in other Indian scripts like Devnagari.

1.7 PROPOSED WORK

In our proposed work we have recognized isolated handwritten characters of Gurmukhi script. In our work we have used some statistical features like projection histograms, distance profiles and zonal density; and one direction feature Background Directional Distribution (BDD) in different combinations to construct different feature vectors. We have presented detailed comparative analysis of the recognition with these feature vectors using three types of classifiers- Support Vector Machines (SVM), K-Nearest Neighbour (K-NN) and Probabilistic Neural Network (PNN). The best result obtained is 95.07% with zonal density and BDD features in combination and SVM classifier for character recognition.We have also recognized Gurmukhi numerals with different features and best recognition rate obtained is 99.2%. The detailed description of the proposed work for character and numeral recognition is given in chapter 3 and chapter 4 respectively.

1.8 OUTLINES OF THESIS REPORT

Chapter 1 is an introductory chapter which includes general and basic overview and introduction of OCR system, overview of Gurmukhi script and introduction and outlines of our proposed work. In chapter 2 we have discussed our literature survey related to earlier approaches regarding character recognition. The detailed description about our proposed work for recognition of Gurmukhi numerals and Gurmukhi characters is presented in chapter 3 and chapter 4 respectively. In our proposed work we have discussed our step by step description of all the phases we have incorporated in our implementation. These steps includes dataset generation, pre-processing, feature extraction and feature vector formation, classification using three types of classifiers (SVM, K-NN and PNN), results evaluation and its analytical discussion and conclusion. Finally we have concluded our proposed work and results in chapter 5.

Chapter 2. LITERATURE SURVEY

2.1 OVERVIEW

In this chapter literature survey regarding the previous work and related approaches about character recognition is presented. The most advanced and efficient OCR systems are designed for English, Chinese and Japanese like scripts and languages. In context to Indian languages and scripts, recently a significant research work is proposed. Most of Indian work on character recognition is dominated by Devnagari script, which is used in writing of Hindi, Marathi, Nepali and Sanskrit languages. Devnagari script is dominating because Hindi which is written in Devnagari script is spoken by a mass of Indian population and is national language of India. After Devnagari, second position of recognition research is taken by Bangla script. Recently much research work is also practiced on other Indian scripts like Gurmukhi, Gujarati, Tamil, Telugu and many other scripts.

In our literature survey we have studied many research approached practiced on many languages and scripts particularly in Indian context. Our emphasis is to study and analyse the feature extraction approaches, observed or reported results and many other relevant issues considerable to any new research work on OCR. After our detailed literature survey we were able to discover the theme of our proposed work including the techniques we have incorporated in various phases, to implement it and to evaluate our results and conclusions.

2.2 WORK RELATED TO CHARACTER RECOGNITION

For Indian languages most of research work is performed on firstly on Devnagari script and secondly on Bangla script. U. Pal and B.B. Chaudhury presented a survey on Indian Script Character Recognition [7]. This paper introduces the properties of Indian scripts and work and methodologies approached to different Indian script. They have presented the study of the work for character recognition on many Indian language scripts including Devnagari, Bangla, Tamil, Oriya, Gurumukhi, Gujarati and Kannada.

U. Pal, Wakabayashi and Kimura also presented comparative study of Devnagari handwritten character recognition using different features and classifiers[23]. They used four sets of features based on curvature and gradient information obtained from binary as well as gray scale images and compared results using 12 different classifiers as concluded the best results 94.94% and 95.19% for features extracted from binary and gray image respectively obtainedwith Mirror Image Learning (MIL) classifier. They also concluded curvature features to use for better results than gradient features for most of classifiers.

A later review of research on Devnagari character recognition is also presented by Vikas Dungre et al. [24]. They have reviewed the techniques available for character recognition. They have introduced image pre-processing techniques for thresholding, skew detection and correction, size normalization and thinning which are used in character recognition. They have also reviewed the feature extraction using Global transformation and series expansion like Fourier transform, Gabor transform, wavelets, moments ; statistical features like zoning, projections, crossings and distances ; and some geometrical and topological features commonly practiced. They also reviewed the classification using template matching, statistical techniques, neural network, SVM and combination of classifiers for better accuracy is practiced for recognition.

Prachi Mukherji and Priti Rege [28] used shape features and fuzzy logic to recognize offline Devnagari character recognition. They segmented the thinned character into strokes using structural features like endpoint, cross-point, junction points, and thinning. They classified the segmented shapes or strokes as left curve, right curve, horizontal stroke, vertical stroke, slanted lines etc. They used tree and fuzzy classifiers and obtained average 86.4% accuracy.

Giorgos Vamvakas et al. [8], [29] have described the statistical and structural features they have used in their approach of Greek handwritten character recognition. The statistical features they have used are zoning, projections and profiling, and crossings and distances. Further through zoning they derived local features like density and direction features. In direction features they used directional histograms of contour and skeleton images. In addition to normal profile features they described in- and out- profiles of contour of images. The structural features they have depicted are end point, crossing point, loop, horizontal and vertical projection histograms, radial histogram, radial out-in and in-out histogram.

Sarbajit Pal et al. [30] have described projection based statistical approach for handwritten character recognition. They proposed four sided projections of characters and projections were smoothed by polygon approximation.

Nozomu Araki et al. [31] proposed a statistical approach for character recognition using Bayesian filter. They reported good recognition performance in spite of simplicity of Bayesian algorithm.

Wang Jin et al. [32] evolved a series of recognition systems by using the virtual reconfigurable architecture-based evolvable hardware. To improve the recognition accuracy of the proposed systems, a statistical pattern recognition-inspired methodology was introduced.

Sandhya Arora et al. [25] used intersection, shadow features, chain code histogram and straight line fitting features and weighted majority voting technique for combining the classification decision obtained from different Multi Layer Perceptron (MLP) based classifier. They obtained 92.80% accuracy results for handwritten Devnagari recognition. They also used chaincode histogram and moment based features in [26] to recognize handwritten Devnagari characters.Chain code was generated by detecting the direction of the next in-line pixel in the scaled contour image. Moment features were extracted from scaled and thinned character image.

Fuzzy directional features are used in [27] in which directional features were derived from the angle between two adjacent curvature points. This approach was used to recognize online handwritten Devnagari characters and result was obtained with upto 96.89% accuracy.Handwritten character recognition using directional features and neural network as classifier is proposed in [34]. In this approach gradient features using Sobel"s mask are extracted and then these features are converted into 12 directional features. Neural network with back propagation is used for classification and 97% accurate result is reported.

Work on Gurmukhi Character Recognition

In particular to Gurumukhi script, Earliest and major contributors founded are Chandan Singh and G. S. Lehal. G. S. Lehal and Chandan Singh proposed Gurumukhi script recognition system [35]. Later they developed a complete machine printed Gurumukhi OCR system [36]. Some of their other research works related to Gurmukhi Script recognition are [37], [38] in which they proposed feature extraction, classification and post processing approaches for Gurumukhi scripts.

Sharma and Lehal proposed an iterative algorithm to segment isolated handwritten words in Gurumukhi script [39]. G.S. Lehal used four classifiers in serial and parallel mode and combined the results of classifiers operated in parallel mode. Combining multiple classifiers their individual weaknesses can be compensated and their strength are preserved.A comparative study of Gurumukhi and Devnagari Script is presented in [40]. In this paper closeness of Devnagari and Gurumukhi script,consonants, conjunct consonants, vowels, numerals and punctuation is discussed.

Anuj Sharma et al. [43], [44] have presented the implementation of three approaches: elastic matching technique, small line segments and HMM based technique, to recognize online handwritten Gurmukhi characters and reported 90.08%, 94.59% and 91.95% recognition accuracies respectively. In elastic maching technique, first they recognised the strokes and then evaluated the character based on the strokes.

For recognition of handwritten Gurmukhi characters following approaches are found in literature survey. Naveen Garg et al. [45] have recognized offline handwritten Gurmukhi characters using neural network and obtained 83.32% average recognition accuracy.Puneet Jhajj et al. [46]used a 48×48 pixels normalized image and created 64 (8×8) zones and used zoning densities of these zones as features. They used SVM and K-NN classifiers and compared the results and observed 72.83% highest accuracy with SVM kernel with RBF kernel. Ubeeka Jain et al. [47]created horizontal and vertical profiles, stored height and width of each character and used neocognitron artificial neural network for feature extraction and classification. They obtained accuracy 92.78% at average. Research articles [48], [49], [50]are publications of our proposed work to recognize handwritten Gurmukhi characters and numerals.

2.3 WORK RELATED TO NUMERAL RECOGNITION

G.S. Lehal and Nivedita Bhatt [41] proposed a system to recognise both Devnagari and English handwritten numerals. They used a set of global and local features, which were derived from the right and left projection profiles of the numeral image. They tested their system on both Devnagari and English numerals independently and found that recognition rate for Devnagari numerals was better. For Devnagari numerals they found recognition rate of 89% and confusion rate of 4.5%. For English set they found recognition rate of 78.4% and confusion rate of 18%.

Reena Bajaj et al. [51] used three different types of features namely density, moment and descriptive component features to recognize the Devnagari numerals. They also used three different neural classifiers and finally the outputs of three classifiers were combined using a connectionist scheme. Thus they obtained classification rate of 89.68% by combining all classifier.

U Bhattacharya and B.B. Chaudhuri [52] proposed majority voting scheme for multi-resolution recognition of hand-printed numerals. They used the features based on wavelet transforms at different resolution levels and multilayer perceptron for classification purpose. They achieved 97.16% recognition rate on a test set of 5000 Bangla numerals. In [53] U. Bhattacharya et al. used ANN and HMM for recognition of handwritten Devnagari numerals. In their proposed scheme they obtained 92.83% recognition rate.

U. Pal et al. [54] proposed a modified quadratic classifier based scheme towards the recognition of off-line handwritten numerals of six popular Indian scripts namely Devnagari, Bangla, Telugu, Oriya, Kannada and Tamil scripts for their experiment. The features used in the classifier were obtained from the directional information of the numerals. They obtained 99.56%, 98.99%, 99.37%, 98.40%, 98.71% and 98.51% accuracy for the scripts respectively.

Sontakke and Patil et al. [55] proposed general fuzzy hyperline segment neural network to recognize handwritten Devnagari numerals. They proposed a rotation, scale and translation invariant algorithm and reported 99.5% recognition rate.

Ramteke and Mehrotra [56] used a method based on invariant moment and the divisions of numeral image for recognition of handwritten Devnagari numerals. They adopted Gaussian Distribution Function for classification and achieved 92% success rate.

Shrivastava and Gharde [57] used Moment Invariant and Affine Moment Invariant techniques for feature extraction and SVM classifier and reported 99.48% recognition rate for handwritten Devnagari Numerals.

Mahesh Jangid and Kartar Singh et al. [58] used a feature extraction technique based on recursive subdivision of the character image to recognize handwritten Devnagari numerals. The character image was subdivided at all iterations such that the resulting sub-image had balanced number of foreground pixels as possible. They achieved 98.98% recognition rate using SVM classifier.

There are also some approaches practiced for recognition of numerals of Indian scripts other than Devnagari. Rajput and Mali [59] recognised isolated Marathi numerals using Fourier descriptors. They used 64 dimensional Fourier descriptors representing the shape of the numerals and invariant to rotation, scale and translation. They used three differentclassifiers namely NN, KNN and SVM independently and achieved 97.05%, 97.04% and 97.85% accuracies respectively.

Apurva Desai [33] used four different profiles of digits namely horizontal, vertical, and left and right diagonal profiles to recognize Gujarati Numerals. He used neural network and achieved 82% success rate.

Work on Gurmukhi Numeral Recognition

D. Sharma and G.S. Lehal et al. recognized Gurumukhi and Roman numeral using 12 structural features comprising of presence of loop, no. of loops, position of loop, entry point of loop(5 features), curve in right upper portion, , presence of right straight line and aspect ratio and statistical features like zoning and directional distance distribution of foreground pixels [42]. They compared the results of different features and observed 92.6% highest accuracy for recognition of Gurmukhi numerals using 256 Directional Distance Distribution (DDD) features. They achieved 95% recognition rate for recognition of Roman numerals.

Ubeeka Jain et al. [47] practiced an approach based neocognitron to recognize Gurmukhi character set including Gurmukhi numerals. They achieved 92.78% recognition rate.

Chapter 3. GURMUKHI CHARACTER RECOGNITION

3.1 OVERVIEW

In this chapter we have proposed the recognition of handwritten Gurmukhi characters. The recognition of Gurmukhi character is our main work and is presented in this chapter. Our extended work to recognize Gurmukhi numerals using proposed recognition technique is presented in chapter 4. In Gurmukhi character recognition, we have used some statistical features like zonal density, projection histograms (horizontal, vertical and both diagonal), distance profiles (from left, right, top and bottom sides). In addition, we have used background directional distribution (BDD) features.

Our database consists of 200 samples of each of basic 35 characters of Gurmukhi script collected from different writers.These samples are pre-processed and normalized to 32×32 sizes. SVM, K-NN and PNN classifiers are used for classification. The performance comparison of features used in different combination with different classifiers is presented and analysed.

To recognize Gurmukhi characters we have used the extracted features in different combinations to form different feature sets. The highest accuracy obtained is 95.07% as 5-fold cross validation of whole database using zonal density and background distribution features by using SVM classifier with RBF kernel.While using K-NN and PNN classifiers we have used highest results on feature set in which background directional distribution features are combined with both diagonal histograms, but these results are lower than the result obtained with SVM classifier. By using K-NN classifier we obtained 86.24% recognition rate and we obtained 87.77% recognition rate while classifying using PNN classifier. These different results obtained are also compared and analysed.

We have also analysed the confusion in classification among similar handwritten patterns of different characters. We have provided the confusion matrices for recognition by considering one division of 5-fold cross validation sets as test dataset.

In the implementation to recognize the characters MATLAB 7.11.0 application software is used in many processing and computation tasks like pre-processing, feature extraction and classification.

3.2 DATASET

In our proposed methodology used to recognize isolated handwritten Gurmukhi characters we have considered 35 basic characters of Gurmukhi alphabet for our experiment. These characters are assumed to bear header line on top. 20 writers of different profiles, age and genders have written these samples in isolated manner on A-4 size white papers. 10 samples of each character by each writer are taken, thus forming a total of 7000 size of our database. The writers were asked to write the characters in such a way that the characters written in each row can be separated from each other on the basis of thoroughly introduced white space horizontally between the character rows. The writers were also asked to introduce white space between characters written in same row. The 35 different characters were arranged to write in different rows and in each row 10 samples of particular character were collected from each writer.The contributors to these data samples were of different educational backgrounds of metric, graduate, post graduate level qualification and different professions as student, teacher, security guard and hostel care-taker.

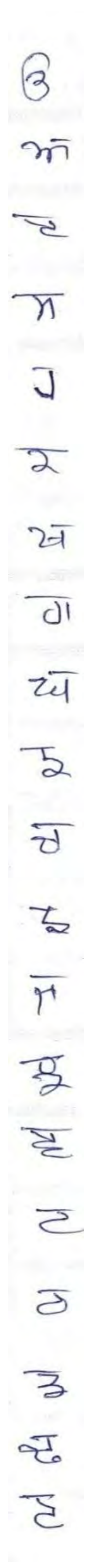
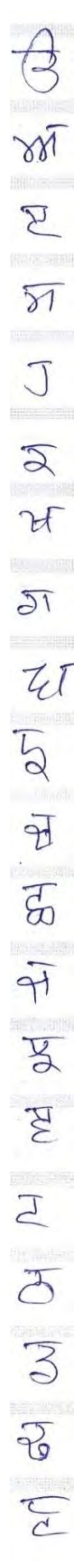
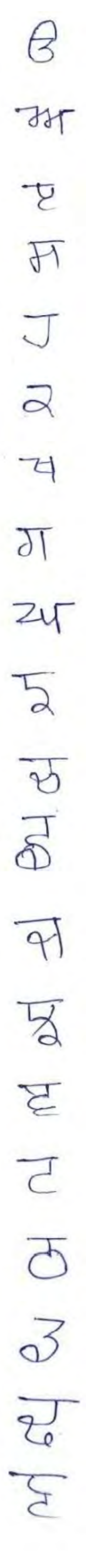
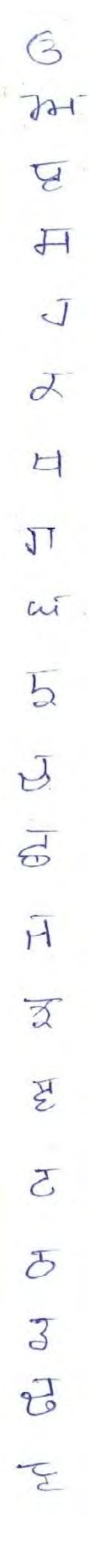
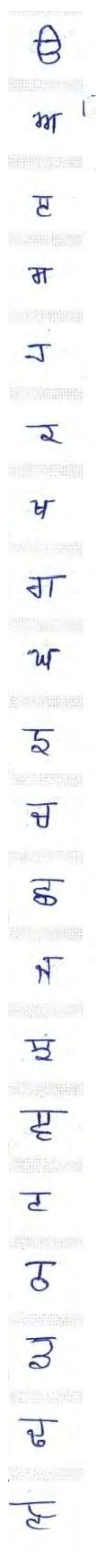
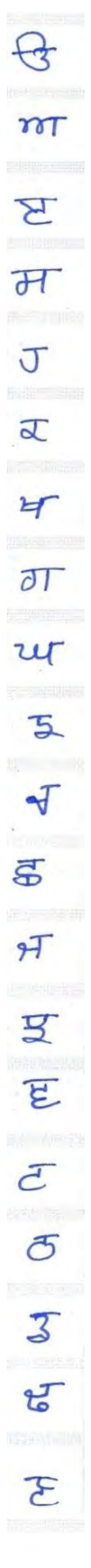
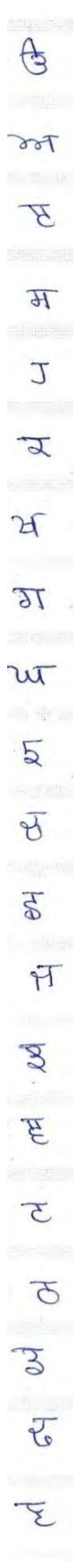
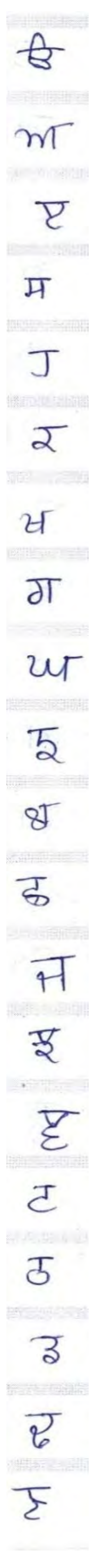
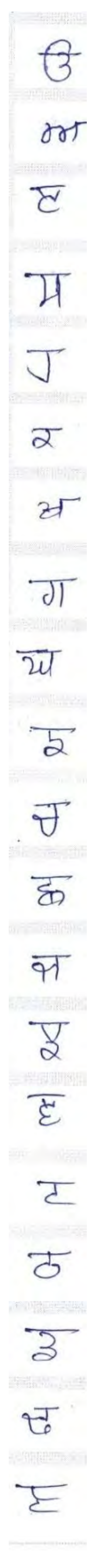
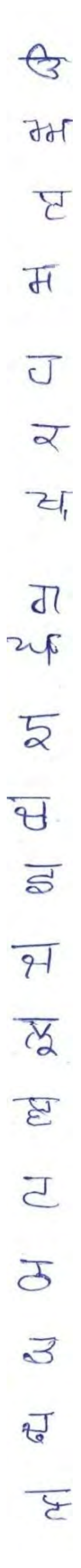
Table 3.1 shows some the samples collected from 10 different writers out of total 20. This table shows single sample out of total 10 samples contributed by these writers for each character.

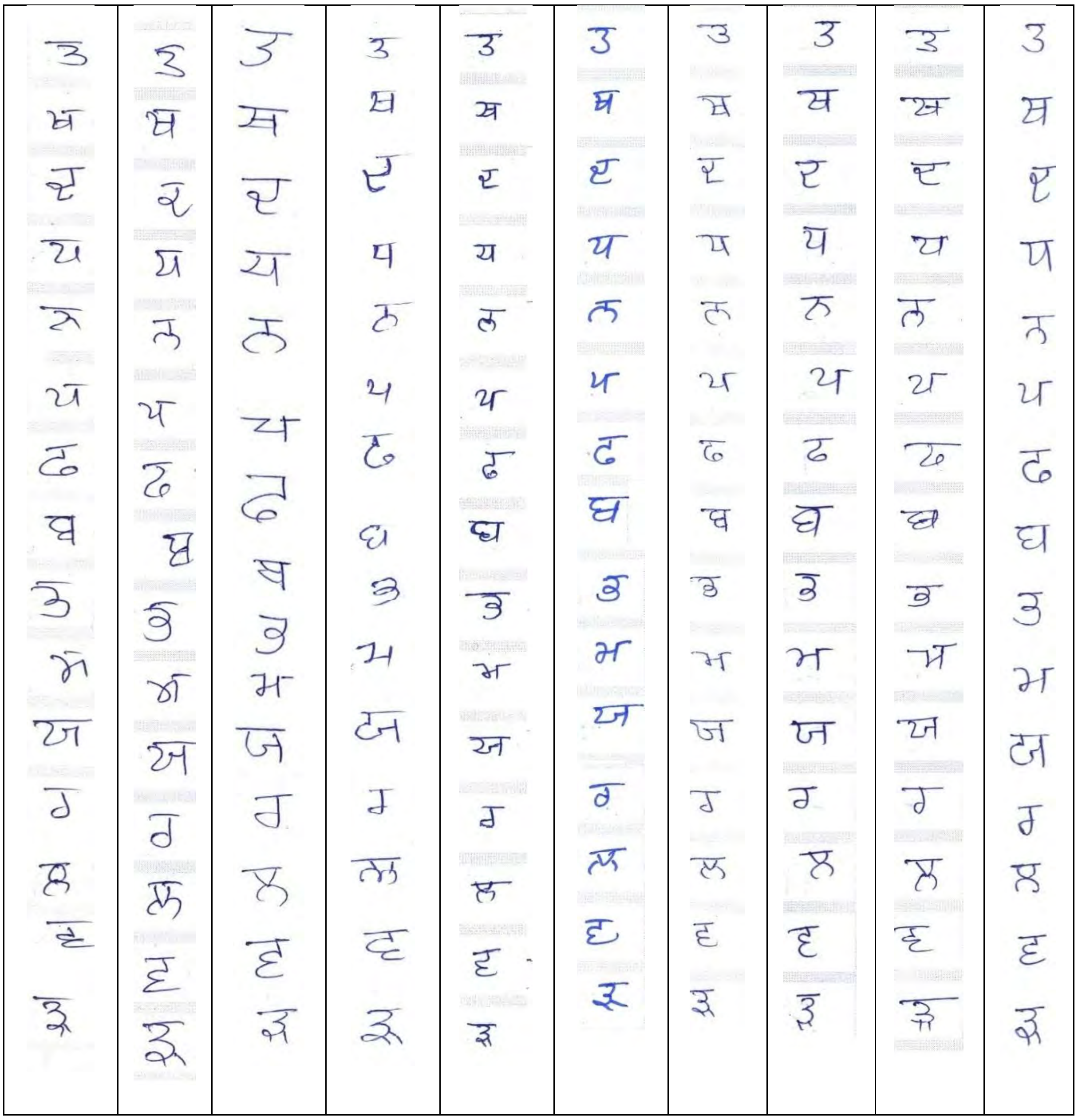
The samples collected from different writers were containing variations in writing styles. The writers wrote the characters in different size using different pens to write. These characters are having different character width. In some characters there are distortions also introduced and amount of such distortion depends on quality of pen ink used to write characters and speed of the writer to write the characters.

3.3 PRE-PROCESSING

The individual characters are segmented and many pre-processing techniques to make these samples easy to extract efficient features are applied. First of all, we scanned handwritten samples in RGB format. Here the pre-processing phase by an example of *ka* (ਕ) character is described. The figure 3.1 shows different stages of pre-processing the character *ka*.

Table 3.1 Samples of Gurmukhi characters from 10 different writers

1	2	3	4	5	6	7	8	9	10
									



Major pre-processing techniques usually approached in image processing are described in section1.3.1. The pre-processing techniques depicted in figure 3.1 and applied in our approach are briefly discussed as follows:

Gray scale image: The pixels in gray scale image show the intensity of corresponding pixels in RGB image. The intensity shows the darkness of image. We have converted coloured images of scanned samples into gray scale images.

Binary image: To convert a gray image into binary image we require to specify threshold value for gray level, dividing or mapping range of gray level into two levels either black or white i.e. either 0 or 1 not between it. We have converted the imaged in binary form using

Otsu‟s threshold method. We have added 0.1 to the threshold value obtained by Otsu‟s method in our approach.

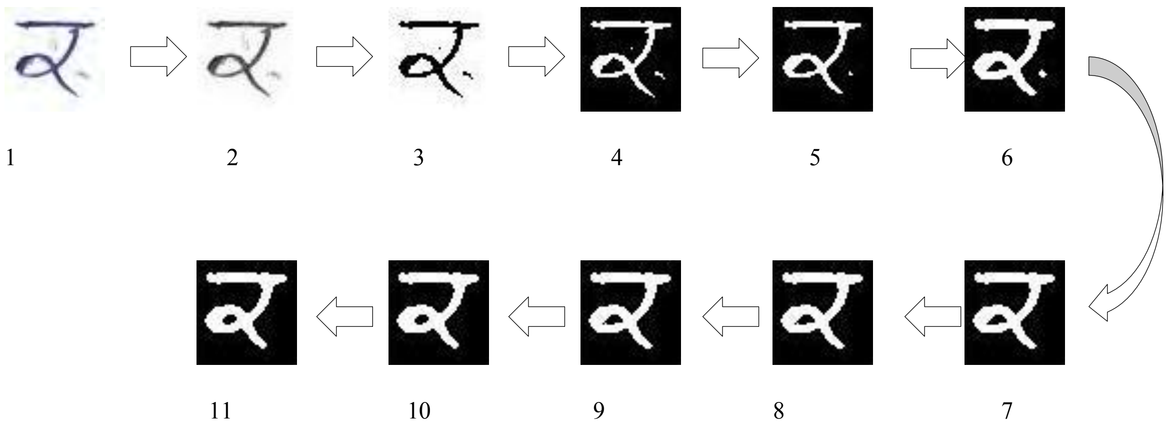


Figure 3.1 Pre-processing stages of character „*ka*‟in the following order of technique: (1) original RGB image, (2) gray image, (3) binary image, (4) image alteration, (5)median filtration, (6) image dilation, (7) removing noise having less than 30 pixels, (8) morphological „bridge‟ operation, (9) morphological „clean‟ operation, (10) morphological „majority‟ operation, (11) morphological „spur‟ operation

Image alteration: The pixels of images are altered from 0 to 1 and vice versa in order to obtain the foreground pixel as 1.

Median filtration: Median filtration on the altered images is applied. Each output pixel is changed to the median value in the 3-by-3 neighbourhood around the correspondingpixel in the input image.

Image dilation: Image dilation to dilate image is applied. The structuring element [0 1 0; 1 1 1; 0 1 0]is used for this purpose.

Removing small objects: Small isolated objects having less than 30 pixels are removed.

Morphological operations: There are some morphological operations which are applied on binary images. Following morphological operationsfor different purposes are applied:

Bridging: It is used to bridge the unconnected pixels in binary image.

Cleaning: It is used to remove isolated pixels (1's surrounded by 0's).

Majority pixel setting: It sets a pixel to 1 if five or more pixels in its3×3 neighbourhood are 1's.

Spurring:It removes end points of lines without removing small objects completely.

Clipping: The image having each isolated character is clipped removing the extra background pixels around the foreground shape of the character image.

Normalization: Then finally we normalized the character image into 32×32 pixels size, which is standard size of each character image used to extract different types of features. However, we also tested other normalization sizes of 48×48 and 40×40 pixels for recognition at starting but eventually we found better results at normalized size of 32×32 pixels. So we decided to use size of 32×32 pixels in our all later experiments.

3.4 FEATURE EXTRACTION

We have used following listed features for our experiment. First three types of features namely zoning density, projection histograms and distance profiles can be categorized as statistical featureswhile fourth type of features (background directional distribution features) can be categorized as directional features. On the basis of these four types of features we have formed 10 feature vectors using different combinations of four basic features. The characters are classified using each of these feature vectors in different classifiers independently.

3.4.1 Zoning Density (ZD) Features

In zoning, the character image is divided into N×M zones. From each zone features are extracted to form the feature vector. The goal of zoning is to obtain the local characteristics instead of global characteristics. We have created 16 (4×4) zones of 8×8 size each out of our 32×32 normalized samples by horizontal and vertical division.(Figure 3.2)

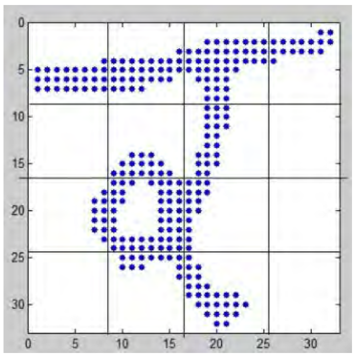


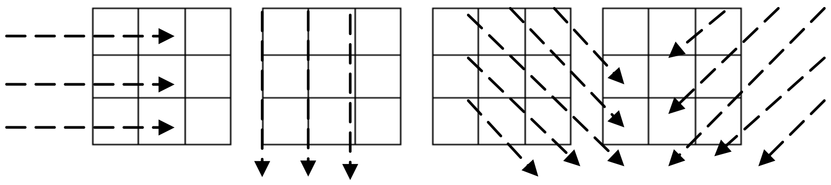
Figure 3.2: Zones of 32×32 normalized handwritten character Gurmukhi character ਕ („*k*‟).

By dividing the number of foreground pixels in each zone by total number of pixels in each zone i.e. 64 we obtained the density of each zone. Thus we obtained 16 zoning density features.

3.4.2 Projection Histogram Features

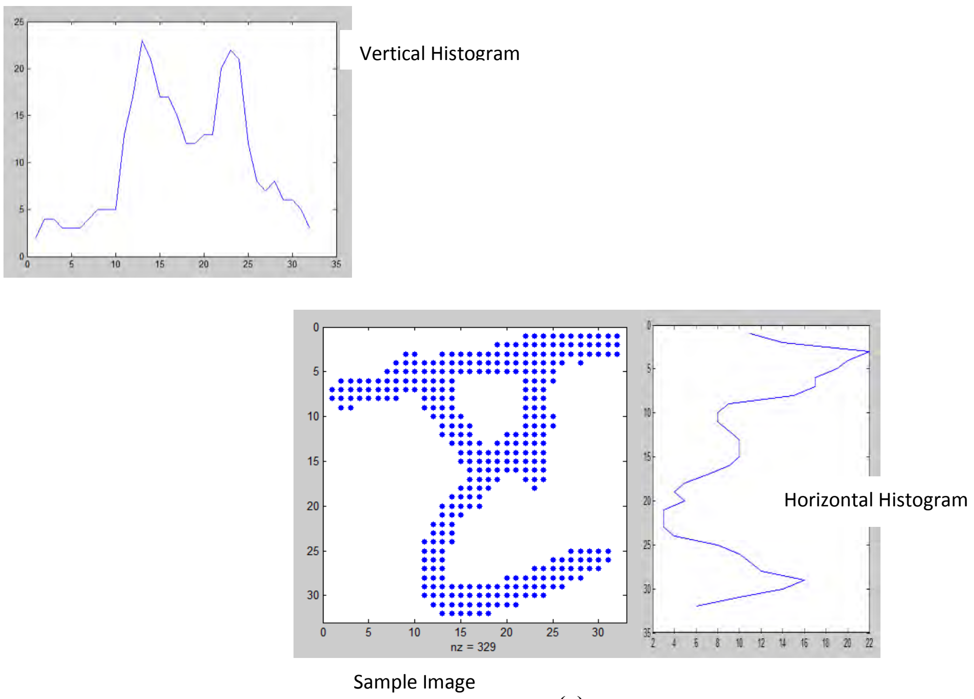
Projection histograms count the number of pixels in specified direction. In our approach we have used three directions of horizontal, vertical and diagonal traversing. We have created three types of projection histograms- horizontal, vertical, diagonal-1 (left diagonal) and diagonal-2 (right diagonal). These projection histograms for a 3×3 pattern are depicted in figure 3.3.

In our approach projection histograms are computed by counting the number of foreground pixels. In horizontal histogram these pixels are counted by row wise i.e. for each row. In vertical histogram the pixels are counted by column wise. In diagonal-1 histogram the pixels are counted by left diagonal wise. In diagonal-2 histogram the pixels are counted by right diagonal wise. The lengths of these features are 32, 32, 63 and 63 respectively according to lines of traversing.

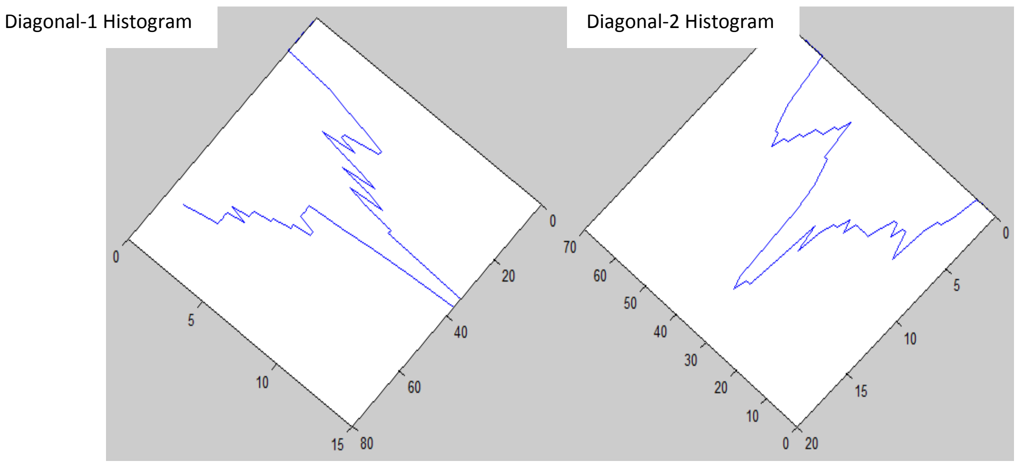


(a) Horizontal Histogram **(b)** Vertical Histogram **(c)** Diagonal-1 Histogram **(d)** Diagonal-2 Histogram

Figure 3.3 Evaluation of 4 types of Projection Histograms on 3×3 patterns



(a)



(b)

Figure 3.4 Four histograms of character „ਝ': **(a)** Sample and its horizontal and vertical histograms, **(b)** diagonal-1 and diagonal-2 histograms of the sample image in (a)

In the figure 3.4 sample image of 32×32 sized Gurmukhi character ਝ („*j*") and its horizontal, vertical, diagonal-1 and diagonal-2 histograms are shown.

3.4.3 Distance Profile Features

Profile counts the number of pixels (distance) from bounding box of character image to outer edge of character. This traced distance can be horizontal, vertical or radial. In our approach we have used profiles of four sides left, right, top and bottom. Left and right profiles are traced by horizontal traversing of distance from left bounding box in forward direction and from right bounding box in backward direction respectively to outer edges of character. Similarly, top and bottom profiles are traced by vertical traversing of distance from top bounding box in downward direction and from bottom bounding box in upward direction respectively to outer edges of character.

We have computed the horizontal distance profiles for each row of pixels and similarly we have computed the vertical profiles for each column of pixels. Hence the size for each of two horizontal and two vertical profiles is 32 and total size of all four profiles is 128.

The figure 3.5 depicts the sample image of character „*gh*" and its four sided profiles. The vertical profile from top is shown at top position of the sample. The vertical profile from bottom direction is shown at bottom position. The horizontal profile from left direction is shown at left side. The horizontal profile from right direction is shown at right position.

The distances of all the profiles are considered from the internal side starting at character sample side.The distance is depicted by the curves

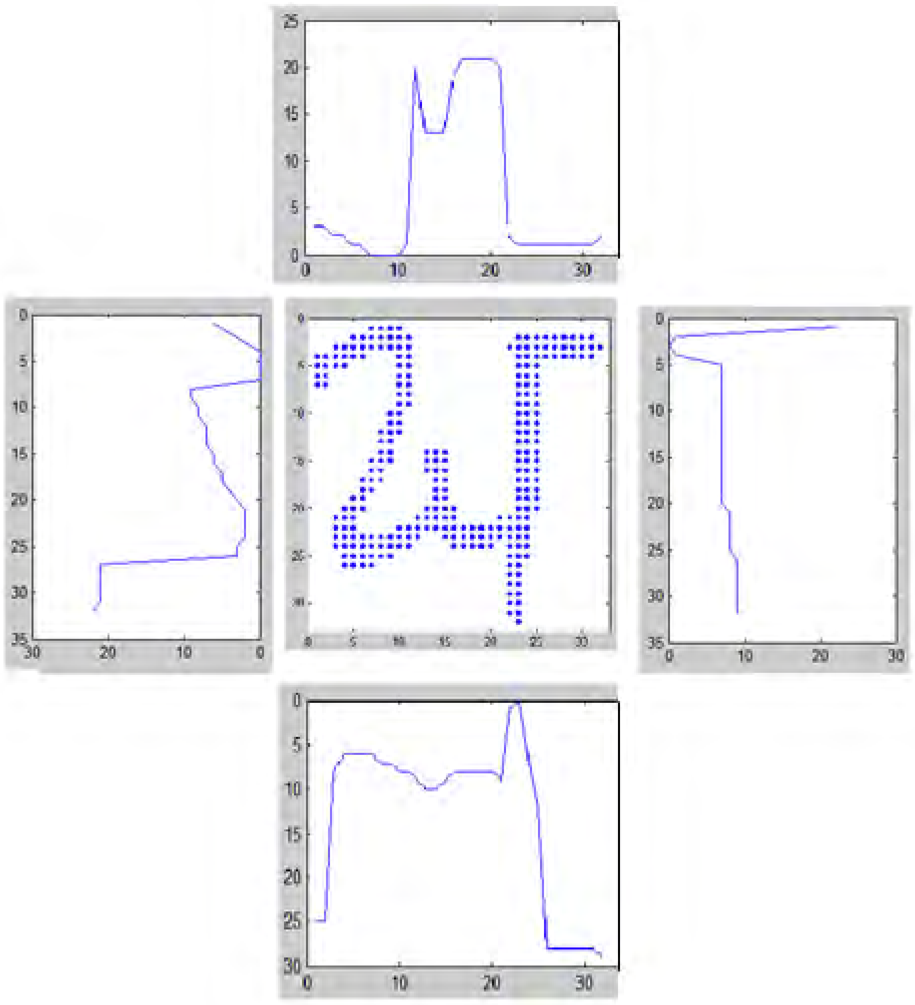
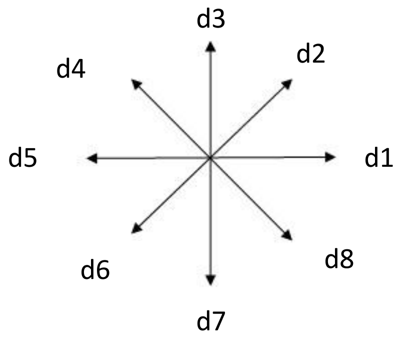


Figure 3.5 A sample image of character ਘ „gh" and its left, right, top and bottom profiles.

3.4.4 Background Directional Distribution (BDD) Features

We have considered the directional distribution of neighboring background pixels of each foreground pixel. We computed 8 directional distribution features. To calculate directional distribution values of background pixels for each foreground pixel, we have used the masks for each direction shown in figure 3.6. The pixel at center „X" is foreground pixel under consideration to calculate directional distribution values of background.

The weight for each direction is computed by using specific mask in particular direction. This weight depicts cumulative weight fraction of background pixels in particular direction. As shown in each mask, the weight for a background pixel coming in middle of each direction is given as 2 while the weight for each of another two pixels coming in both sides is given as 1.



(a)

0	0	1
0	X	2
0	0	1

d1

0	1	2
0	X	1
0	0	0

d2

1	2	1
0	X	0
0	0	0

d3

2	1	0
1	X	0
0	0	0

d4

1	0	0
2	X	0
1	0	0

d5

0	0	0
1	X	0
2	1	0

d6

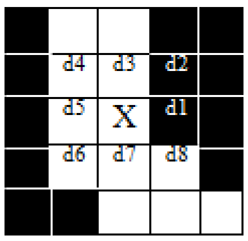
0	0	0
0	X	0
1	2	1

d7

0	0	0
0	X	1
0	1	2

d8

(b)



(c)

Figure 3.6:BDD features computation- directions, masks and example: **(a)** 8 directions used to compute directional distribution, **(b)** Masks used to compute directional distribution in different directions. **(c)** An example of sample

To illustrate the computation of BDD features, we have to compute directional distribution value for foreground pixel „X" in direction d1for the sample given in figure 3.6(c). We need to superimpose the mask for d1 direction on the sample image coinciding the centered pixel X. The sum of mask values of d1 mask coinciding to background pixels neighbouring to X in figure 3.6(c) i.e. d1 and d2 (i.e. 1+2) will be feature value in direction d1. Similarly we obtained all directional distribution values for each foreground pixel. Then, we summed up all similar directional distribution values for all pixels in each zone. Zones are described earlier in zoning density features description. Thus we finally computed 8 directional distribution feature values for each zone comprising total 128 (8×16) values for a sample image.

Following algorithm describes the computation of BDD features for each character image of size 32×32 pixels having 4×4 zones and thus each zone having 8×8 pixels size. Eight directional features D1, D2, . . . D8 are computed for each zone.

Algorithm for computation of BDD features

```
// image size=32×32, number of zones =16 // (4×4), pixels in each
zone=64(8×8)
 (1)  START
 (2)  Initialize D1,D2,... D8 = 0;
 (3)  Repeat step 4 for each zone Z(r,c)
      //where, r=1,2,3,4; c=1,2,3,4;
 (4) Repeat step 5 for each pixel P(m,n) of the zone Z(r,c)
     //where, m=1,2,...8; n=1,2,...8;
 (5) IF pixel P(m,n) is foreground
      Then, 8 distribution features (d1,d2 …d8) for zone Z(r,c) are
      computed by summing up the mask values specified for neighbouring
      background pixels in particular direction (see fig 2(b)) as follows:
        D1 = D1 + 2× ~P(m,n+1) + ~P(m-1,n+1) + P(m+1,n+1);
        D2 = D2 + 2× ~P(m-1,n+1)+ ~P(m-1,n) + ~P(m,n+1);
        D3 = D3 + 2× ~P(m-1,n)+ ~P(m-1,n-1) + ~P(m-1,n+1);
        D4 = D4 + 2× ~P(m-1,n-1)+ ~P(m,n-1) + ~P(m-1,n);
        D5 = d5 + 2× ~P(m,n-1) + ~P(m-1,n-1) + ~P(m+1,n-1);
        D6 = d6 + 2× ~P(m+1,n-1)+ ~P(m,n-1) + ~P(m+1,n);
        D7 = d7 + 2× ~P(m+1,n) + ~P(m+1,n-1) + ~P(m+1,n+1);
        D8 = d8 + 2× ~P(m+1,n+1)+ ~P(m+1,n) + ~P(m,n+1);
(6)  STOP
```

3.4.5 Formation of Feature Vectors

To form feature vectors to be used in classification we have derived10 different feature vectors using different combinations of above described four types of basic features.

Table 3.210 sets of feature vectors formed with different combinations of features

Feature Vector	Included Features	Size
FV1	Zonal Density (ZD)	16
FV2	Profiles	128
FV3	Histograms	190
FV4	BDD	128
FV5	Profiles + ZD	144
FV6	BDD + ZD	144
FV7	Profiles + horizontal and vertical histograms(HVH)	192
FV8	BDD + HVH	192
FV9	BDD + Profiles	256
FV10	BDD + Diagonal (both) histograms	254

These different sets of feature vectors are shown in table 3.2. It can be observed, FV1 to FV4 each consists of single type of features while FV5 to FV10 consist of different combinations of the features FV1 to FV4.

3.5 CLASSIFICATION

We have used three types of classifiers in our implementation- SVM, K-NN and P-NN. Among these SVM is popular and efficient and also produced most efficient results in our implementation. In Gurmukhi character recognition we have used all the classifiers while only SVM classifier is used to recognize Gurmukhi numerals.

5-fold Cross Validation: To obtain the results using the different classifiers we have used 5-fold cross validation. In V-fold cross validation we first divide the training set into V equal subsets. Then one subset is used to test by classifier trained by other remaining V-1 subsets. By cross validation each sample of train data is predicted and it gives the percentage of correctly recognized dataset.

In our implementation,first we created randomly generated 5-fold cross-validation index of the same length as the total number of samples i.e. 7000. This index contains equal proportions of the integers 1 through 5. These integers are used to define a partition of whole dataset into 5 disjoint subsets each with different index integer. We have used one division for testing and remaining divisions for training. We did so 5 times, each time changing the testing dataset to different division and considering remaining divisions for training. Thus we got 5 sets of feature vectors containing training and testing dataset in the size ratio of 4:1.

The average recognition accuracy of these randomly generated 5 sets of training and testing is referred as cross validation accuracy.

3.5.1 Support Vector Machines (SVM) Classifier

At present SVM is popular classification tool used for pattern recognition and other classification purposes. Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. The standard SVM classifier takes the set of input data and predicts to classify them in one of the only two distinct classes. SVM classifier is trained by a given set of training data and a model is prepared to classify test data based upon this model.For multiclass classification problem, we decompose multiclass problem into multiple binary class problems, and we design suitable combined multiple binary SVM classifiers.

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. The figure 3.7 shows the classification of objects having class one of the two: either *triangle* or *diamond*. The separating line defines a boundary on the right side of which all objects are diamond and to the left of which all objects are triangle. Any new object falling to the right is labelled, i.e., classified, as diamond (or classified as triangle if it falls to the left of the separating line).

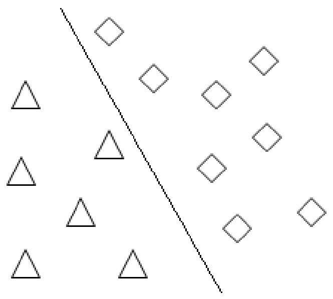


Figure 3.7 Linear classification of objects by SVM into two classes

The above is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective classes (diamond and triangle in this case) with a line. Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). This complex separation is represented by a complex function rather than linear function and such classification of objects is shown in figure 3.8(a). Compared to the previous schematic, it is clear that a full separation of the diamond and triangle objects would require a curve (which is more complex than a line).

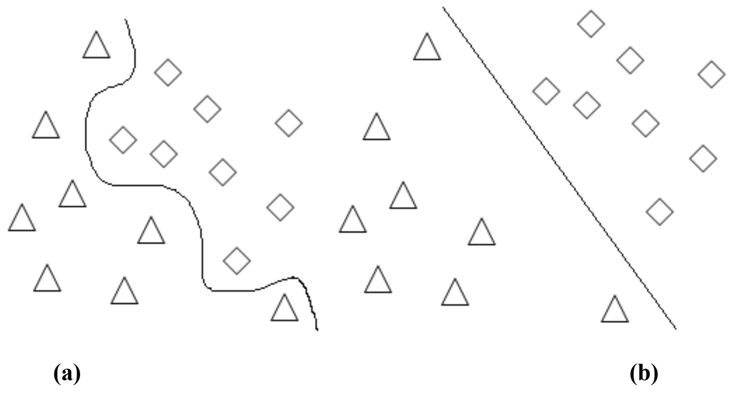


Figure 3.8 Mapping of SVM classification from complex to linear: **(a)** classification of objects by a complex kernel function (separating curve) **(b)** mapping of input space of figure (a) into feature space making objects linearly separable

Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. Support Vector Machines are particularly suited to handle such tasks.

The basic motivation behind SVM is to map (transform) the input space of original objects (shown in figure 3.8(a)) into a feature space making the mapped objects linearly separable. This mapping is done by a set of mathematical functions, known as kernels. All we have to do in SVM is to find the optimal line that that can saperate the objects.

Given a training set of instance-label pairs(x_i, y_i) , $i = 1,2, \dots l$ where $x_i \in R^n$, i.e. having n independent variables or features for a particular instance and $y \in \{1, -1\}^l$, i.e. class label either 1 or -1 for each sample among total l instances. The support vector machines (SVM) require the solution of the following optimization problem, i.e. minimization of error function:

$$\min_{w,b,\xi} \frac{1}{2} W^T W + C \sum_{i=1}^{l} \xi_i \,,$$

subject to the constraints:

$$y_i(W^T \phi(x_i) + b) \geq 1 - \xi_i \,,$$

and, $\xi_i \geq 0$.

where C is the capacity constant, W is the vector of coefficients, b a constant and $\xi_i$are parameters for handling non-separable data (inputs). The index i labels the N training cases or instances. Here $y_i \in \pm 1$ are the class labels and x_i are the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space. Here training vector $x_i$are mapped into a higher (may be infinite) dimensional space by the function ϕ . SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term.

Kernels of SVM Classifiers

In addition to basic linear kernel, many other kernels are also proposed by researchers. Following are the four basic kernels used in SVM classifications:

Linear: $K(x_i, x_j) = x_i^T . x_j$.

Polynomial: $K(x_i, x_j) = (\gamma x_i^T . x_j + r)^d , \gamma > 0$.

Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$.

Sigmoid: $K(x_i, x_j) = \tanh(\gamma {x_i}^T x_j + r)$.

Here $\gamma, r,$ and d are kernel parameters. Among these kernels RBF kernel is mostly used. We have also used RBF kernel in our approach.

RBF kernel nonlinearly mapssamples into a higher dimensional space so it, unlike the linear kernel, can handle thecase when the relation between class labels and attributes is nonlinear. Furthermore,the linear kernel is a special case of RBF since the linearkernel with a certain value of penalty parameter C has the same performance as the RBF kernel with some parameters (C, γ). In addition, the sigmoid kernel behaves like RBF for certain parameters.

Another reason for preferring RBF is the number of hyperparameters which influences the complexityof model selection. The polynomial kernel has more hyperparameters thanthe RBF kernel.

Finally, the RBF kernel has fewer numerical difficulties. One key point is $0 < K_{ij} \leq 1$ in contrast to polynomial kernels of which kernel values may go to infinity($\gamma {x_i}^T x_j + r > 1$) or zero ($\gamma {x_i}^T x_j + r < 1$) while the degree is large. The sigmoid kernel is also not valid under some parameters.

In our implementation we have used a multiclass SVM classifier. Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements.The dominating approach for doing so is to reduce the single multiclass problem into multiple binary classification problems.We obtained such multiclass SVM classifier tool LIBSVM [60].A practical guide for SVM and its implementation is available at [61].More description about SVM classifier can be found at [18], [19] and [20].

The steps followed for recognition using SVM classifier are below:

- Transforming data to the format of an SVM package
- Conducting simple scaling on the data
- Considering the RBF kernel $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$
- Using cross-validation to find the best parameter C and γ
- Using the best parameter C and γobtained to train the whole training set
- Testing

We transformed the feature vector into $M \times N$ matrix, where M is number of samples and N is number of features for each sample. All these features are scaled in range [0,1].

We have used 5-fold cross validation for obtaining training and testing data sets and observed the results by testing the test dataset after training the classifier with training dataset.

Parameter Selection: The effectiveness of SVM depends on the selection of kernel, the kernel's parameters, and soft margin parameter C. Best combination of C and γ is often selected by a grid-search with exponentially growing sequences of C and γ, for example, $C \in \{2^{-5}, 2^{-4}, \ldots\ 2^{14}, 2^{15}\}and\gamma \in \{2^{-15}, 2^{-14}, \ldots\ 2^{2}, 2^{3}\}$.Each combination of parameter choices is checked using 5-fold cross validation, and the parameters with best cross-validation accuracy are picked. The final model, which is used for testing is then trained on the whole training set using the selected parameters

3.5.2 K-Nearest Neighbour (K-NN) Classifier

K-NN classifier uses the instance based learning by relating unknown pattern to the known according to some distance or some other similarity function. It classifies the object by majority vote of its neighbour. Because it considers only neighbour object to a particular level, it uses local approximation of distance function. It means lazy or instance learning is used in K-NN while in other classifiers as SVM eager learning is used. K specifies the number of nearest neighbours to be considered and the class of majority of these neighbours is determined as the class of unknown pattern. In the figure 3.9 the classification of objects with k=3 (solid line circle) and k=5 (dotted circle) is shown.

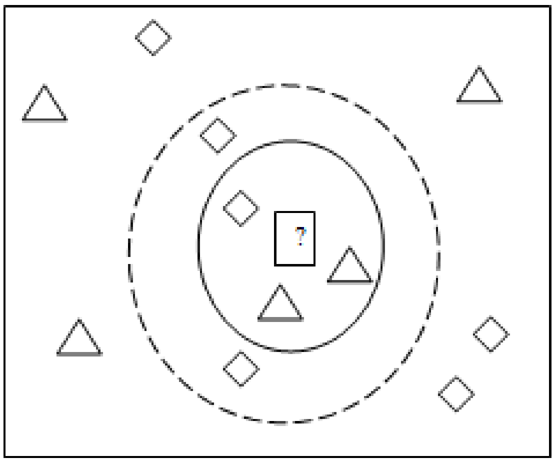


Figure 3.9 Classification of objects using K-NN with k=3 and k=5

It is notable that class of object is altered in both cases. At k=3 the unknown sample will be classified as triangular shape object while at k=5 it will be classified as diamond shape object because at k=3 triangular objects are in majority (within inner solid circle) while at k=5 diamond shaped objects are in majority (within outer dashed circle).

The distance function used to find the nearest neighbour are: Euclidian distance, sum of absolute differences, cosine, correlation and percentage of bits that differ.

Parameter Selection: More robust models can be achieved by locating k, where k > 1, neighbours and letting the majority vote decide the outcome of the class labelling. A higher value of k results in a smoother, less locally sensitive, function. Generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct.The best choice of k depends upon the data. A good k can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbour algorithm.

More description about K-NN classifier can be found at [21] and [22].

In our implementation we have used Euclidean distance as the distance parameter, while using other distance functions we obtained degraded results. The same data format used in SVM is used in K-NN. We have tested test set of data after training with training set and also have used 5-fold cross validation.

3.5.3 Probabilistic Neural Network (PNN) Classifier

Probabilistic Neural Network is a special type of Neural Network classifiers. It is a multi-layered feed-forward neural network classifier in which known probability density function (pdf) of the population is used to classify unknown patterns. PNN is closely related to Parzen window pdf estimator.

PNN consists of four layered neural network, Figure 3.104 shows a typical PNN model.

PNN receives the n-dimensional feature vectors $x\ (x_1, x_2, \dots\ x_n)$ as input and in first layer these inputs are applied to the input neurons $x_i (1 \leq i \leq n)$ and passed to the neurons in the first hidden layer (i.e. second layer). This second layer consists of the probability density functions (pdf); mostly Gaussian functions are used as pdf which are formed using the given set of data points. m_kGaussians are computed for each class where $1 \leq k \leq c$. The third layer performs an average operationof the outputs from the second layer for eachclass.The fourth layer performs a vote, selecting thelargest value. The associated class label isthen determined.

If the probability density function (pdf) ofeach population is known, then anunknown, X, belongs to class “i” if:$f_i(x) > f_j(x)$, for all $j \neq i$, here $f_k(x)$ is the pdf of unknown x for class k.

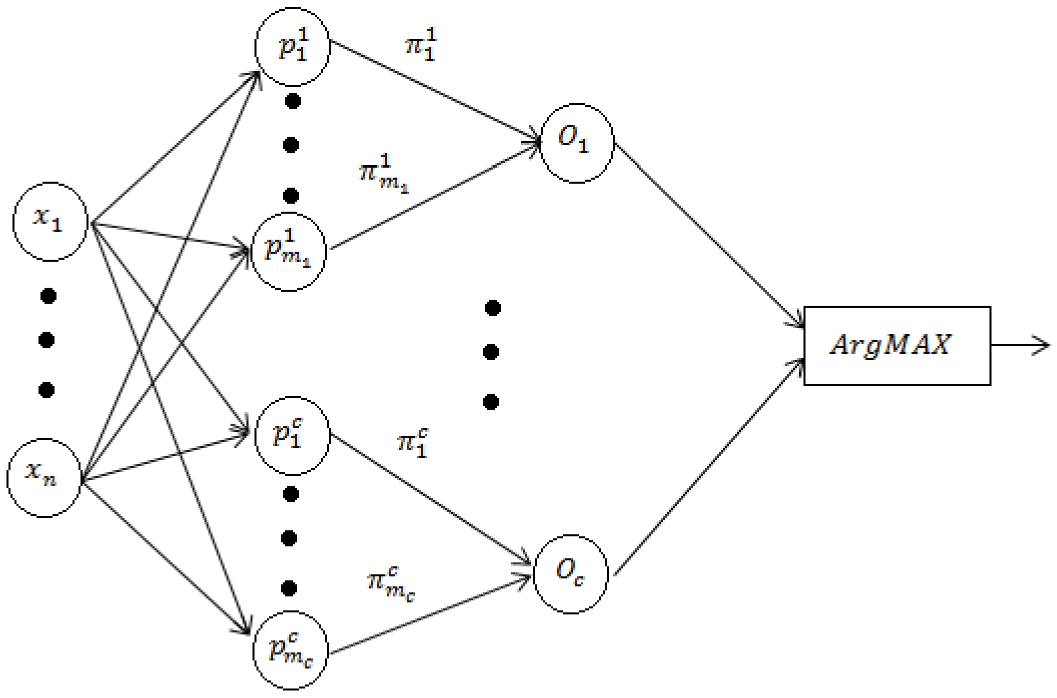


Figure 3.10 Multi-layered PNN classification model

Probability Density Function (pdf):pdf is estimated by using the samples ofthe populations (i.e. the training set). pdf for a population of n samples is computed by computing the average of pdfs of all samples:

$$p(x) = \frac{1}{n\sigma}\sum_{k=1}^{n} W\left(\frac{x - x_k}{\sigma}\right),$$

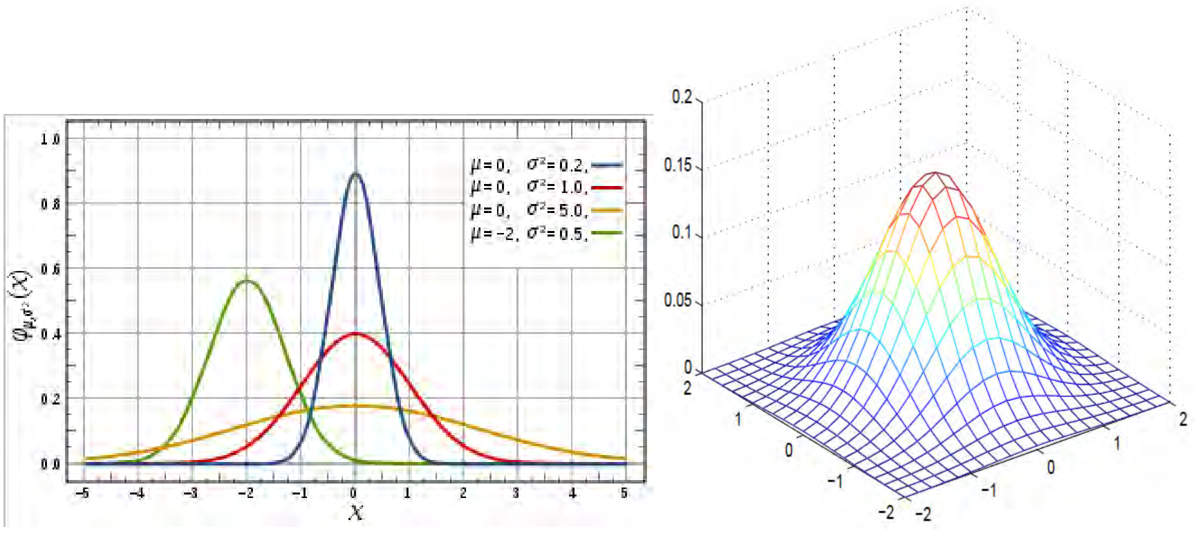
where, $x_k$is k^{th} sample, x is unknown input, W is weighting function also called window function or kernel and σ is smoothing parameter in the form of standard deviation. The estimated pdf approaches the truepdf as the training set size increases, aslong as the true pdf is smooth. This is also called Parzen‟s pdf estimator. We cangeneralize the idea and allow the use of otherwindow functions so as to yield other Parzenwindow density estimation methods.

The Gaussian pdf is a popular kernel for Parzen-window density estimation, being infinitely differentiable and thereby lending the same property to the Parzen-window pdf estimate $p(x)$. The Parzen-window estimate with the Gaussian kernel becomes (for 1-D):

$$p(x) = \frac{1}{n\sigma\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right)$$

In d-dimentional space the pdf with Gaussian kernel will become:

$$p(x) = \frac{1}{n(\sigma\sqrt{2\pi}\,)^d} \sum_{i=1}^{n} \exp\left(-\frac{(x - x_k)^2}{2\sigma^2}\right)$$



(a) (b)

Figure 3.11 Examples of 1-D and 2-D Gaussian pdf :(a) 1-D Gaussian pdf , **(b)** 2-D Gaussian pdf

In figure 3.11 1-D and 2-D Gaussian pdf are shown.

In our implementation we have used Parzen PNN tool. More detailed description about Parzen window probability density estimation can be found in [14], [15] and description about PNN classifier can be found at [16], [17].In Parzen PNN tool $N \times M$ feature vector is used where M is number samples (instances) and N is number of features of each instance. That is, in this classifier feature used in SVM and K-NN is transposed first before classification.

3.6 RESULTS AND ANALYSIS

To obtain the recognition results we have used 5-fold cross validation with the three classifiers. First we created randomly generated 5-fold cross-validation index of the length of size of dataset. This index contains equal proportions of the integers 1 through 5. These integers are used to define a partition of whole dataset into 5 disjoint subsets. We used one division for testing and remaining divisions for training. We did so 5 times, each time changing the testing dataset to different division and considering remaining divisions for training. Thus we got 5 sets of feature vectors containing training and testing dataset in the size ratio of 4:1.

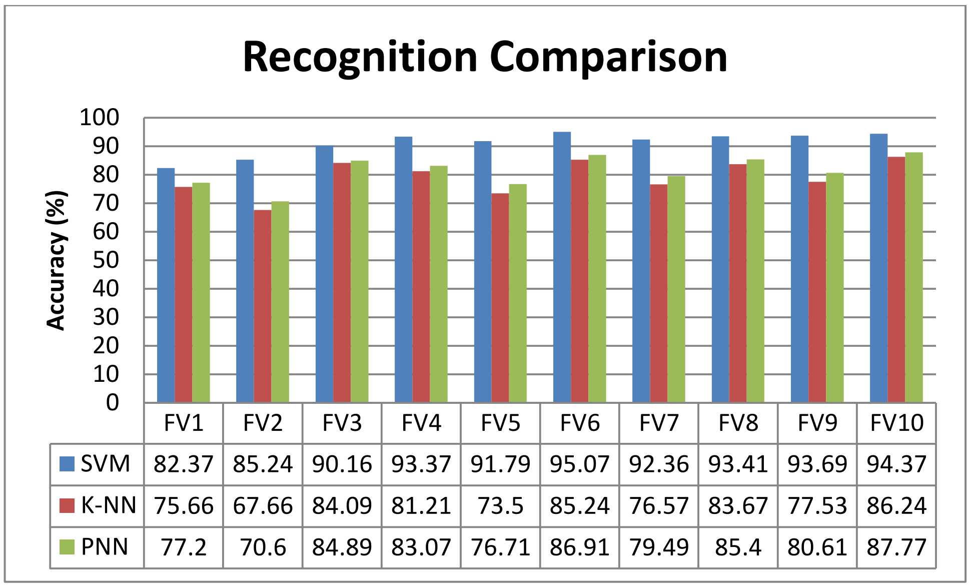
The average recognition accuracy of these randomly generated 5 sets of training and testing is referred as cross validation accuracy. In following sections recognition results obtained by character and numeral recognition are discussed.

In the table 3.3 5-fold cross-validation results using SVM, K-NN and PNN classifiers are shown. These are the highest results with selected feature set and classifier. The corresponding parameters of these classifiers, at which these optimized results are obtained, are also shown. Features used in these feature vectors are shown in table 3.2. For selection of these parameters to obtain optimized results, first we used small sample of whole dataset and observed the parameters giving highest results. Later we refined this optimization by cross validation of whole dataset. The variation of the results with different parameters and hence refinement of the parameters giving optimum results for particular classifier is described in next subsections of results with each classifiers by considering some or all feature vectors.

Table 3.3 Optimized Cross validation Results with different classifiers and corresponding parameters

Feature Vector (size)	**SVM**			**K-NN**		**PNN**	
	C	**γ**	**Accuracy (%)**	**K**	**Accuracy (%)**	**σ**	**Accuracy (%)**
FV1 (16)	6	1.8	82.3714	3	75.6571	0.15	77.2000
FV2 (128)	4	0.22	85.2429	5	67.6571	0.34	70.6000
FV3 (190)	4	0.20	90.1571	5	84.0857	0.27	84.8857
FV4 (128)	22	0.22	93.3714	1	81.2143	0.23	83.0714
FV5 (144)	22	0.07	91.7857	5	73.5000	0.31	76.7143
FV6 (144)	>16	0.28	95.0714	1	85.2429	0.22	86.9571
FV7 (192)	11.3	0.05	92.3571	7	76.5714	0.36	79.4857
FV8 (192)	6	0.20	93.4143	5	83.6714	0.30	85.4000
FV9 (256)	45	0.07	93.6857	3	77.5286	0.34	80.6143
FV10 (254)	>16	0.28	94.3714	1	86.2429	0.26	87.7857

The highest result obtained is 95.07% with SVM as classifier and FV6 as feature set. It can be observed that results for all the feature sets with SVM are superior comparative to K-NN and PNN. K-NN is giving the lowest results comparative to SVM and PNN for all feature vectors. The best results obtained with PNN and K-NNare observed at FV6 and FV10 feature sets. However the results with FV10 are slightly higher due to increased number of features. But this increased number of features has not shown any proportionality with results in the case of SVM. Figure 3.12 shows the comparison of all the results obtained with ten feature vectors and three classifiers.



	FV1	FV2	FV3	FV4	FV5	FV6	FV7	FV8	FV9	FV10
🟦 SVM	82.37	85.24	90.16	93.37	91.79	95.07	92.36	93.41	93.69	94.37
🟥 K-NN	75.66	67.66	84.09	81.21	73.5	85.24	76.57	83.67	77.53	86.24
🟩 PNN	77.2	70.6	84.89	83.07	76.71	86.91	79.49	85.4	80.61	87.77

Figure 3.12 Result comparison with different feature vectors and three classifiers

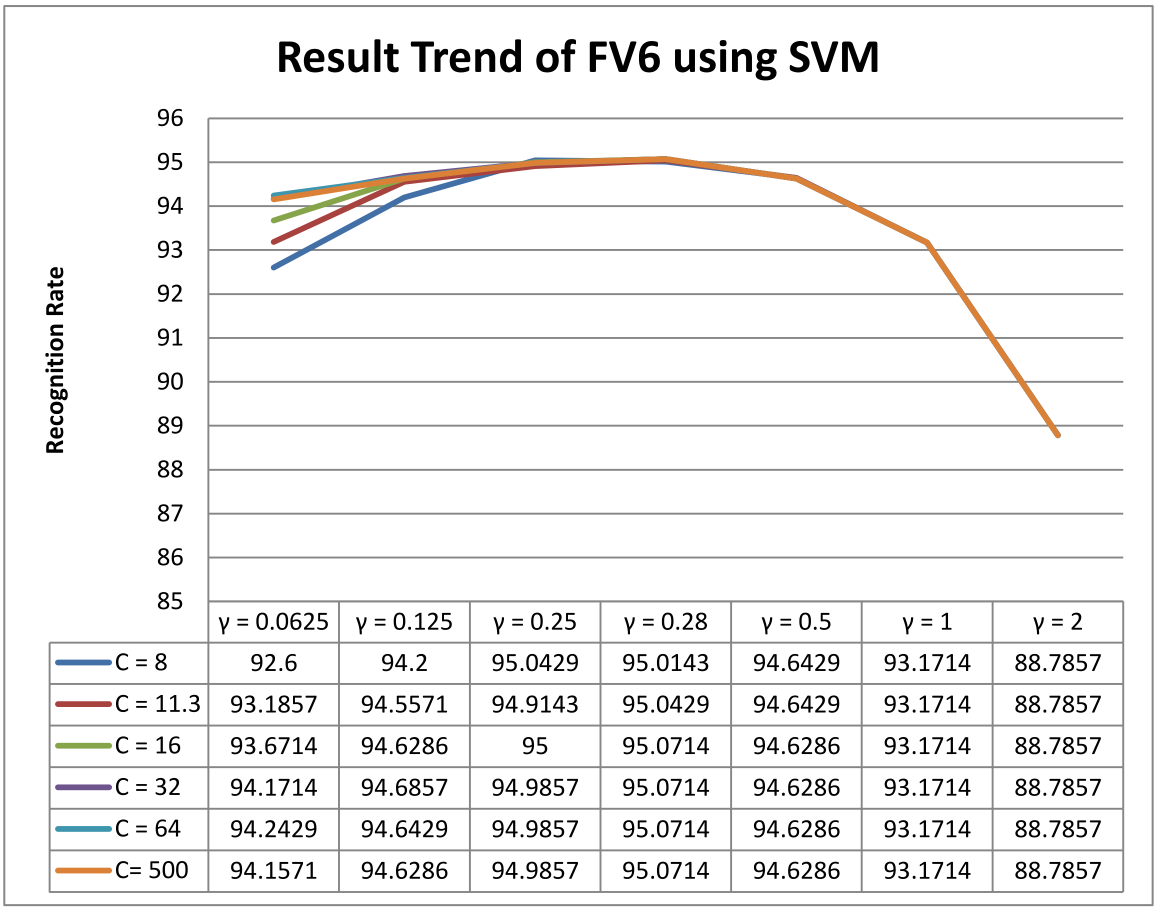
3.6.1 Results with SVM Classifier

As shown in table 3.3, the results obtained with SVM are superior comparative to both PNN and K-NN. The optimized results shown in the table were obtained after refinement of the optimum results for each feature set and classifier. For illustration, the variation trend of the results using SVM with FV6 at changed value of γ with different values of C is shown in figure 3.13.

By observation of figure 3.13 it can be analyzed that the results vary significantly only at small values of C in combination with small values of γ. These results are more sensitive to change with parameter $\gamma$of RBF kernel comparative to C. At larger values of C results are stable and variation is negligible. In figure 3.13 the significant difference is only at smaller values of γ (0.06 to 0.25) and even these variations are only observable at smaller values of C (8 to 16) and at larger values of C (>16) all the results are mostly coinciding for all range of γ. Most of the results of SVM listed are observed at larger range of C tested upto 500.It is clear from the table shown below the figure that highest result of our proposed work is obtained for

FV6 feature set. This highest result is 95.07% which is obtained at γ = 0.28 with combination of C = 16, 32, 64 and 500 (i.e. C ≥ 16).

On variation in γ it can be observed that for initial values there are low results and at larger values there is a significant decrement in the recognition rates at all the values of C.



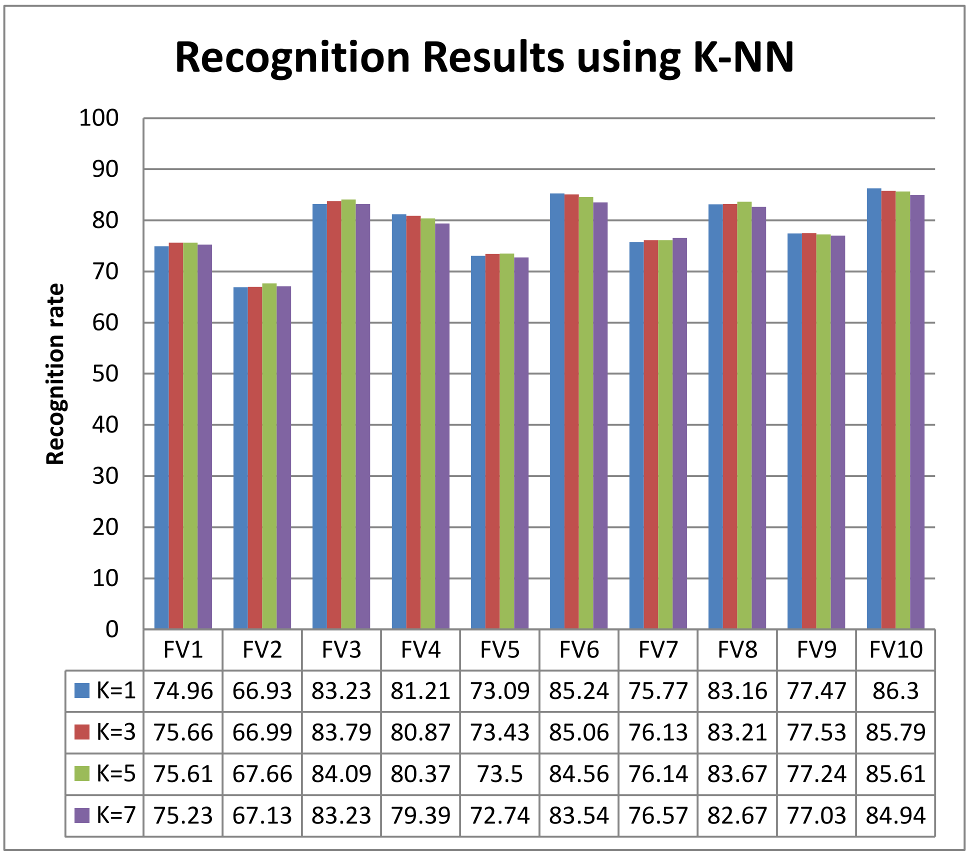
	γ = 0.0625	γ = 0.125	γ = 0.25	γ = 0.28	γ = 0.5	γ = 1	γ = 2
▬ C = 8	92.6	94.2	95.0429	95.0143	94.6429	93.1714	88.7857
▬ C = 11.3	93.1857	94.5571	94.9143	95.0429	94.6429	93.1714	88.7857
▬ C = 16	93.6714	94.6286	95	95.0714	94.6286	93.1714	88.7857
▬ C = 32	94.1714	94.6857	94.9857	95.0714	94.6286	93.1714	88.7857
▬ C = 64	94.2429	94.6429	94.9857	95.0714	94.6286	93.1714	88.7857
▬ C= 500	94.1571	94.6286	94.9857	95.0714	94.6286	93.1714	88.7857

Figure 3.13 Result Trends of FV6 using SVM

3.6.2 Results with K-NN Classifier

The K-NN results are obtained by using „Euclidian‟ distance parameter, while the results on „correlation‟ parameter were observed to be lower in all cases. The values for k parameter with highest obtained results are listed in table 3.3. The trend of the results with K-NN with K=1,3,5,7 for all 10 feature sets is depicted in figure 3.14.

It is clear from table 3.3 that there is no strong relation on the fact that at which particular value of K the highest result has to observe. Different feature sets are showing highest results at different values of K. In K-NN best results are obtained with FV10 and second best results are obtained with FV6. Both of these feature sets are comprised of BDD features combined with diagonal histograms and zonal density respectively. These feature sets have 254 and 144 number of features respectively. Hence the increased recognition rate of FV10 is obtained at the cost of significantly increased features. The classification using K-NN is less time consuming because it is instant based learning methodology instead of eager learning used in SVM.



	FV1	FV2	FV3	FV4	FV5	FV6	FV7	FV8	FV9	FV10
🟦 K=1	74.96	66.93	83.23	81.21	73.09	85.24	75.77	83.16	77.47	86.3
🟥 K=3	75.66	66.99	83.79	80.87	73.43	85.06	76.13	83.21	77.53	85.79
🟩 K=5	75.61	67.66	84.09	80.37	73.5	84.56	76.14	83.67	77.24	85.61
🟪 K=7	75.23	67.13	83.23	79.39	72.74	83.54	76.57	82.67	77.03	84.94

Figure 3.14 Recognition results using K-NN for all feature sets at K=1,3,5,7.

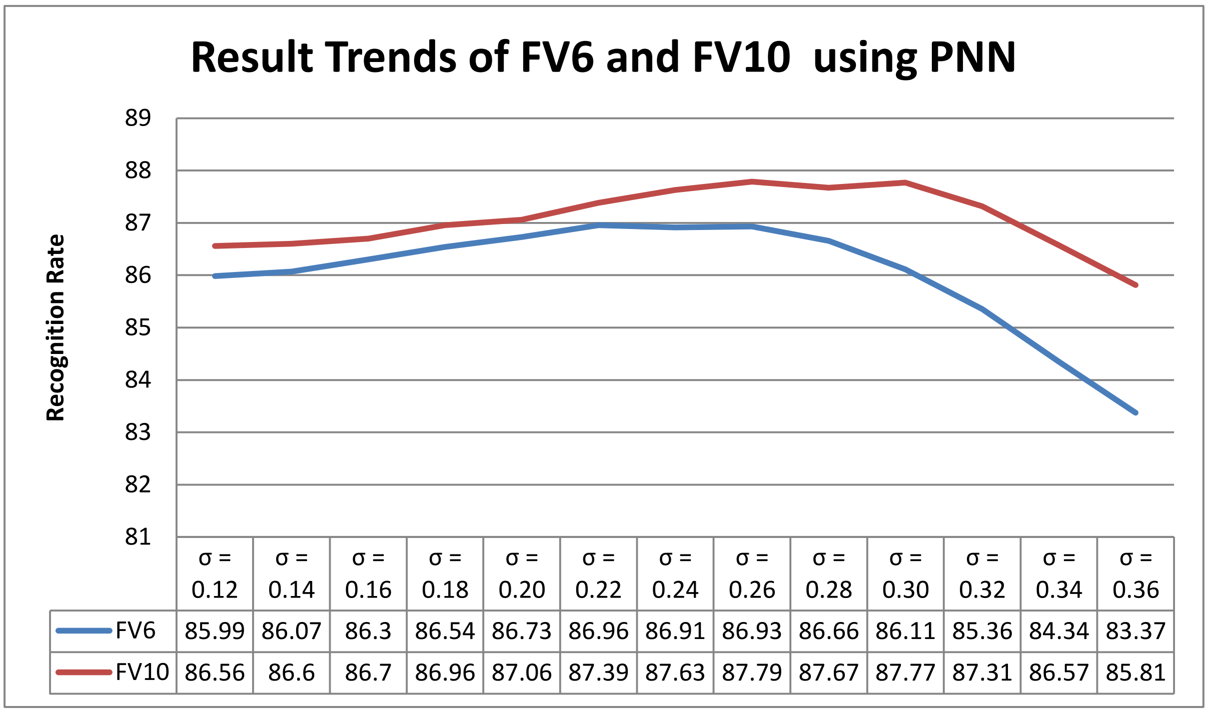
3.6.3 Results with PNN Classifier

The highest result obtained with all the feature sets using PNN are listed in table 3.3. The optimized values of smoothing parameter σ at which these results are obtained are also listed in the table. The parameter σ was refined for each feature set to obtain the best result. For illustration, the result trends with variation of σ from 0.12 to 0.36 are shown in figure 3.15 for FV6 and FV10 given below.

It can be observed that with FV6 the highest result obtained is 86.96% observed at σ = 0.22 and with FV10 the highest result observed is 87.79% observed atσ = 0.26.

It can be observed from the figure 3.15 that there is a comparatively quick decrement in recognition after a certain value of σ.

The results obtained with PNN are lower than SVM but better than K-NN. The comparison of results with all three classifiers is listed in table 3.3.



	σ = 0.12	σ = 0.14	σ = 0.16	σ = 0.18	σ = 0.20	σ = 0.22	σ = 0.24	σ = 0.26	σ = 0.28	σ = 0.30	σ = 0.32	σ = 0.34	σ = 0.36
FV6	85.99	86.07	86.3	86.54	86.73	86.96	86.91	86.93	86.66	86.11	85.36	84.34	83.37
FV10	86.56	86.6	86.7	86.96	87.06	87.39	87.63	87.79	87.67	87.77	87.31	86.57	85.81

Figure 3.15 Result trends of FV6 and FV10 using PNN

3.6.4 Comparison with Earlier Approaches

Naveen Garg et al. [45]have recognized offline handwritten Gurmukhi characters using neural network and obtained 83.32% average recognition accuracy.

Anuj Sharma in his Ph.D. thesis [44] has recognized the online handwritten Gurmukhi characters using three approaches. In first approach [43] online Gurmukhi characters are recognized using elastic matching algorithms by matching unknown stroke against the strokes database giving 90.08% recognition accuracy. The second approach, called small line segments, is based on elastic matching and chaincode algorithm and gives 94.59% recognition accuracy. The third approach which is based on hidden Markov model, gives 91.95% accuracy.

For offline handwritten Gurmukhi character recognition two more approaches are reported, one is proposed by Puneet Jhajj et al. [46] and another one by Ubeeka Jain et al. [47]. In both approaches work is done on isolated characters. In Puneet Jhajj et al.‟s approach zoning density features were used and the results obtained with two types of classifiers SVM and K-NN were compared. They observed the best result as 73.83% with SVM classifier using RBF kernel. In Ubeeka Jain et al.‟s approach neocognitron neural network was used for isolated

Gurmukhi character recognition. They reported 92.78% accuracy average to both known and unknown writers.

The table 3.4 depicts the comparison of our proposed approach used to recognize Gurmukhi characters with all these previous approaches.

Table 3.4 Comparison of Gurmukhi character recognition with earlier approaches

Proposed by	Features used	Classifier	Accuracy
Naveen Garg et al. [45]	Structural Features	Neural Network	83.32%
Anuj Sharma et al. [43]	Strokes recognition and matching	Elastic matching	90.08%
Anuj Sharma et al. [44]	Small line segments	Elastic Matching	94.59%
Anuj Sharma et al. [44]	HMM elements	Hidden Markov model	91.95%
Puneet Jhajj et al. [46]	Zoning density (64)	SVM with RBF kernel	73.83%
Ubeeka Jain et al. [47]	Profiles, width, height, aspect ratio, neocognitron	Neocognitron Neural Network	92.78%
Our proposed approach	Zoning density and background directional distribution features	SVM with RBF kernel	95.07%

3.6.5 Character Confusion Matrix

In character recognition of any scripts the similar patterns are generally confused. The features we have selected for classification cannot distinguish such patterns. This situation becomes worst in case of handwritten patterns. The handwritten patterns have different style variations and sometimes different characters are written similarly. In printed characters there minute differences in the structures of similar looking characters, but generally these distinguishing differences in handwriting are ignored,

We can opt out to classify most confusing characters separately in later stage after identifying the characters with major amount of confusion by using specific features.

Table 3.5 shows the confusionmatrix of Gurmukhi characters recognized using SVM classifier. This confusion matrix is constructed from recognition results of one of the 5 sets of training and testing used in 5-fold cross validation. This ਖ table gives 95% recognition rate. The most confusions observed in the table are: ਖwith ਘ, ਪ, ਮ; ਛwith ਙ, ਵ ; ਜwith ਚ, ਲ; ਡwith ਚ, ਤ, ਰ; ਢwith ਚ; ਥwith ਖ, ਬ, ਰ; ਦwith ਕ, ਚ, ਰetc. these confusions cause the decreased recognition performance.

Table 3.5 Confusion Matrix of Gurmukhi Characters classified with SVM

Sr. No.	Character	No. of Samples	Recognized	Misclassified with	Accuracy (%)
1	ੳ	35	35	NIL	100.00
2	ਅ	43	42	1(ਜ)	97.67
3	ੲ	35	35	NIL	100.00
4	ਸ	43	41	1(ਗ), 1(ਲ)	91.11
5	ਹ	42	41	1(ਕ)	97.62
6	ਕ	42	40	1(ਬ), 1(ੜ)	95.24
7	ਖ	44	40	1(ਘ), 1(ਪ), 3(ਮ)	88.89
8	ਗ	38	37	1(ਸ)	97.37
9	ਘ	40	37	1(ਅ), 2(ਪ)	92.5
10	ਙ	31	31	NIL	100
11	ਚ	39	38	1(ਥ)	97.44
12	ਛ	45	43	1(ਙ), 1(ਞ)	95.56
13	ਜ	36	33	2(ਚ), 1(ਲ)	91.67
14	ਝ	43	42	1(ਣ)	97.67
15	ਞ	45	42	3(ਵ)	93.33
16	ਟ	36	35	1(ਦ)	97.22
17	ਠ	35	35	NIL	100.00
18	ਡ	28	25	1(ਚ), 1(ਤ), 1(ਰ)	89.29
19	ਢ	42	37	5(ਚ)	88.10
20	ਣ	46	44	1(ਹ), 1(ਙ)	95.65
21	ਤ	32	31	1(ਭ)	96.88
22	ਥ	45	39	1(ਖ), 4(ਬ), 1(ਰ)	86.67
23	ਦ	42	38	1(ਹ), 1(ਕ), 1(ਚ), 1(ਰ)	88.37
24	ਧ	37	36	1(ਸ)	97.30
25	ਨ	41	40	1(ਠ),	97.56
26	ਪ	43	43	NIL	100.00
27	ਫ	40	39	1(ਢ)	97.50
28	ਬ	45	40	5(ਥ),	88.89
29	ਭ	41	39	1(ੳ), 1(ਲ)	95.12
30	ਮ	35	34	1(ਖ)	97.14
31	ਯ	47	44	1(ਗ), 1(ਧ), 1(ਬ)	93.62
32	ਰ	33	31	1(ਚ), 1(ਡ)	93.94
33	ਲ	44	41	1(ਸ), 2(ਥ)	93.18
34	ਵ	46	45	1(ਭ)	97.83
35	ੜ	37	37	NIL	100.00
Total		**1400**	**1330**	**70**	**95.00**

3.6.6 Result Conclusions

Conclusion aboutFeatures: The best recognition rate obtained for recognizing the Gurmukhi character is obtained with feature vector FV6 i.e. feature vector consisting of zonal density and BDD features. This highest recognition accuracy for Gurmukhi characters is obtained with SVM classifier and is observed as 95.07%. The results obtained with other feature vectors are shown in table 3.3. Our best result obtained is also compared with some available earlier approaches of Gurmukhi character recognition. Both the features used in FV6 are zone based, i.e. extraction of local features is more useful than global features used in other feature vectors. Out of these local features Background directional distribution features have major contribution to gain optimal recognition results. It is because these features consist of directional as well as local features. The combination of profile and histogram features with BDD (in FV8, FV9, and FV10) is comparatively less relevant to gain best results. Additionally these combinations of features have a drawback of increased size of feature vector, increasing the computing complexity.

Conclusion about Classifier: Among three classifiers SVM classifier shows the best performance to recognize Gurmukhi characters as well as Gurmukhi numerals. In most of earlier approaches of character recognition also SVM performed better. The other two classifiers have shown a poor recognition performance in comparison to SVM classifier. The results obtained with PNN are inferior than SVM but still better than K-NN. In all listed feature vectors, best results are obtained with SVM classifier, then secondly by PNN classifier and K-NN produced lowest results comparatively.

Chapter 4. GURMUKHI NUMERAL RECOGNITION

4.1 OVERVIEW

In this chapter our extended work to recognize Gurmukhi numerals is described. The same technique described in chapter 3 is implemented here to recognize the numerals. The dataset of Gurmukhi numerals for our implementation is collected from 15 different persons. Each writer contributed to write 10 samples of each of numeral of 10 different Gurmukhi digits.We have used three different feature sets.First feature set is comprised of distance profiles having 128 features. Second feature set is comprised of different types of projection histograms having 190 features. Third feature set is comprised of zonal density and Background Directional Distribution (BDD) forming 144 features.The SVM classifier with RBF (Radial Basis Function) kernel is used for classification. PNN and K-NN classifiers are also used. We have obtained the highest 5-fold cross validation accuracy using SVM as 99.2% using second feature set consisting of 190 projection histogram features. On third and first feature sets recognition rates 99.13% and 98% are observed. The results with other classifiers are also presented.

4.2 NUMERAL DATA SET

We have collected the Gurmukhi numeral dataset from 15 different persons. Each writer contributed to write 10 samples of each of numeral of 10 different Gurmukhi digits. These samples are taken on white papers written in an isolated manner. In the table 4.1 Gurmukhi numerals and 10 different sets of samples of each are shown. These samples are showing the variations of different writing styles. These variations include differences in size, strokes, width and distortions of various samples.

4.3 GURMUKHI NUMERAL RECOGNITION PROCESS

We have adopted here three sets of features, and SVM classifiers in the recognition process. The short and specific description of these phases is given here.

4.3.1 Pre-processing

We have applied many pre-processing techniques like gray scaling, binarization, image alteration, median filtration, image dilation, noise removal, some morphological operations, clipping and normalization. All these pre-processing techniques are described in 3.3.

Table 4.1 Gurmukhi numerals and its handwritten samples

	0	1	2	3	4	5	6	7	8	9
1	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
2	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
3	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
4	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
5	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
6	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
7	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
8	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
9	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯
10	੦	੧	੨	੩	੪	੫	੬	੭	੮	੯

4.3.2 Feature Extraction

We have used following three sets of features extracted to recognize Gurmukhi numerals. These approaches are adopted from our earlier practice used to recognize isolated Gurmukhi handwritten characters described in Chapter 3.

1. Distance Profile Features
2. Projection Histogram Features
3. Zonal density and Background Directional Distribution (BDD) Features

Distance Profile Features: Distance profile features we have adopted as first feature set are described in 3.4.3. We have computed and extracted the features similarly in the context of Gurmukhi numerals. We have used four types of profiles left, right, bottom and top each of size 32 corresponding to each pixel in row and column. Thus total size of profile features is 128.

Projection Histogram Features: Projection histogram features are described in 3.4.2. There are horizontal and vertical histogram features of size 32 of each. There are left and right diagonal histogram features of size 63 each. Thus total size of four types of histogram features is 190.

Zonal Density and Background Directional Distribution Features: Zonal density features are explained in 3.4.1 and Background Directional Distribution (BDD) features are explained in 3.4.4. Zonal density features are of size 16 and BDD features are of size 128. We combined these two types of features to form third feature set of size 144.

4.3.3 Classification

We have classified the samples of Gurmukhi numerals using three different sets of features independently. We have used SVM, PNN and K-NN classifiers to recognize the Gurmukhi numerals.

4.4 RESULTS AND ANALYSIS

To observe the results, we have used 5-fold cross validation and obtained accuracy as 99.2% with SVM classifier using second feature set consisting of 190 projection histogram features. On third and first feature sets recognition rates 99.13% and 98% are observed.

The table 4.2 depicts the optimized results obtained with different features set at refined parameters using SVM. The result variation is more sensitive to value γ comparison to C.

The highest results obtained with PNN using three feature sets FV1, FV2 and FV3 is shown in table 4.3.From the table 4.3 it is clear that with PNN the highest result is obtained with FV3 as 98.33%. The highest results observed with the three feature sets (FV1, FV2 and FV3) are listed in table 4.4 with corresponding parameter K. From the table 4 it can be observed that the highest result with K-NN is obtained using FV3 feature set as 98.51%. The obtained with FV1 and FV2 are 94.58% and 97.72% respectively.

Table 4.2 Gurmukhi numeral Recognition results with SVM

Feature Set	Recognition Rate	Parameters
Distance Profiles (128) [FV1]	98%	C=16; γ = 0.03- 0.15
Projection Histograms (190) [FV2]	99.2%	C=16; γ = 0.05-0.15
Zonal density and BDD (144) [FV3]	99.13%	C=16; γ = 0.05-0.55

Table 4.3 Gurmukhi numeral recognition results with PNN

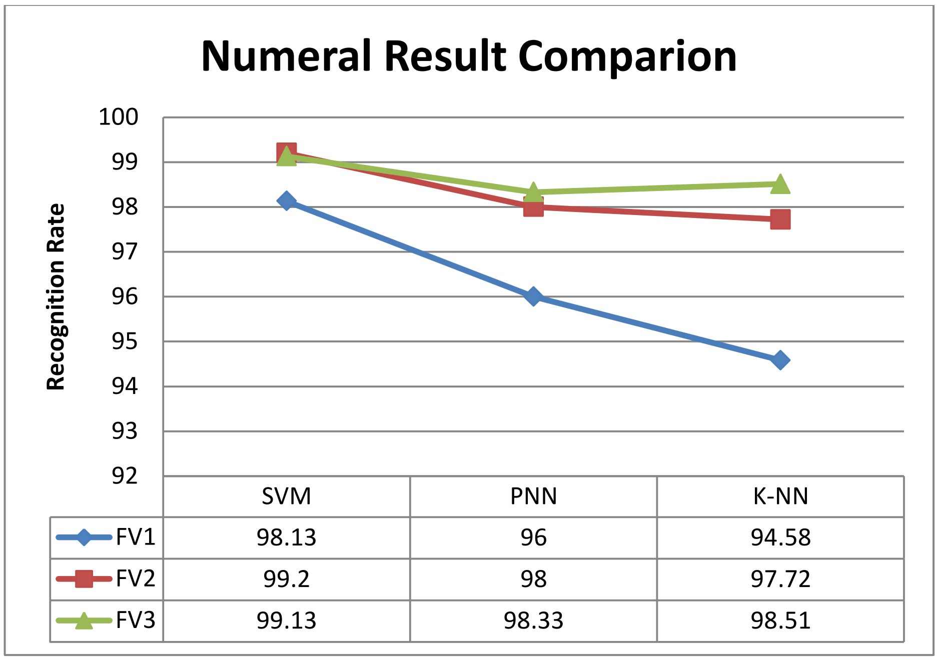
Feature Set	Recognition Rate	Parameters
FV1	96%	σ = 0.30
FV2	98%	σ = 0.30
FV3	98.33%	σ = 0.15, 0.20

Table 4.4 Gurmukhi numeral recognition results with K-NN

Feature Set	Recognition Rate	Parameters
FV1	94.58%	K = 3
FV2	97.72%	K = 1
FV3	98.51%	K = 1

While recognising with SVM and observing the results at other values of parameter C, it is analysed that increasing the value of C irrespective of any change in γ slightly decreases the recognition rate, but after a certain increment normally after 64 at higher values of C the recognition rate becomes stable. In contrast, the recognition rate always changes with the change in γ. The optimized results are obtained at C=16 and γ value in range 2^{-5} to 2^{-1}.

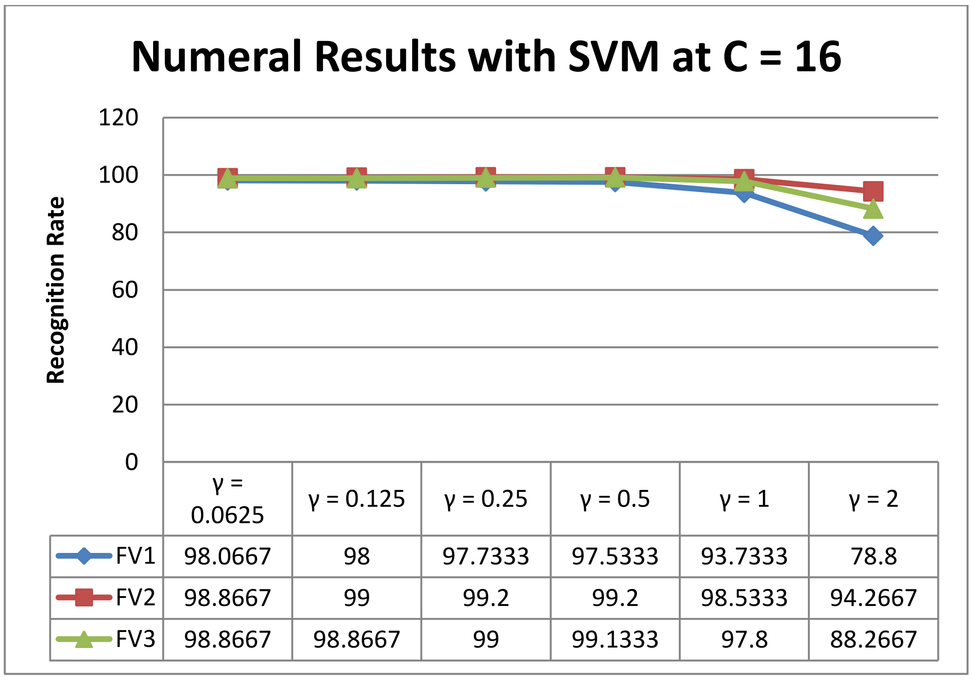
In the figure 4.1 all the highest results obtained with three classifiers using each of feature set are compared. In the case of PNN and K-NN the highest results are obtained with FV3, But in case of SVM the highest result is obtained using FV2. This result is also highest among all the results with all the classifiers. All the results with SVM are highest comparative to other classifiers- PNN and K-NN.



	SVM	PNN	K-NN
FV1	98.13	96	94.58
FV2	99.2	98	97.72
FV3	99.13	98.33	98.51

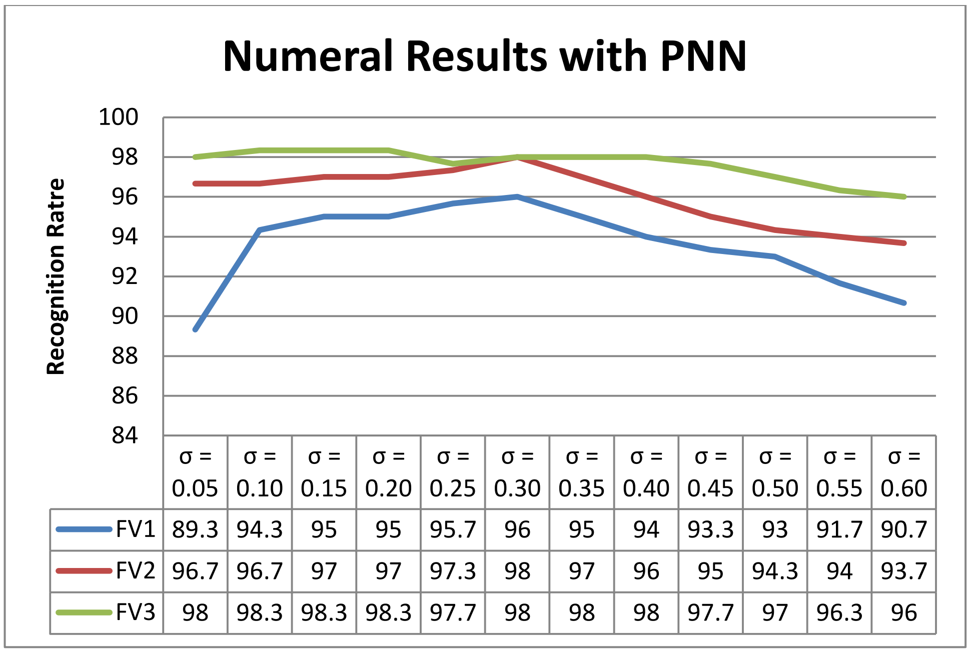
Figure 4.1 Comparison of Numeral results with the three classifiers and different feature sets

The trends of result variation with SVM classifier at C = 16 and varying γ is shown in figure 4.2. The various results with PNN classifier at different σ is shown in figure 4.3. Similarly the results with K-NN at different values of parameter K are shown in figure 4.4.



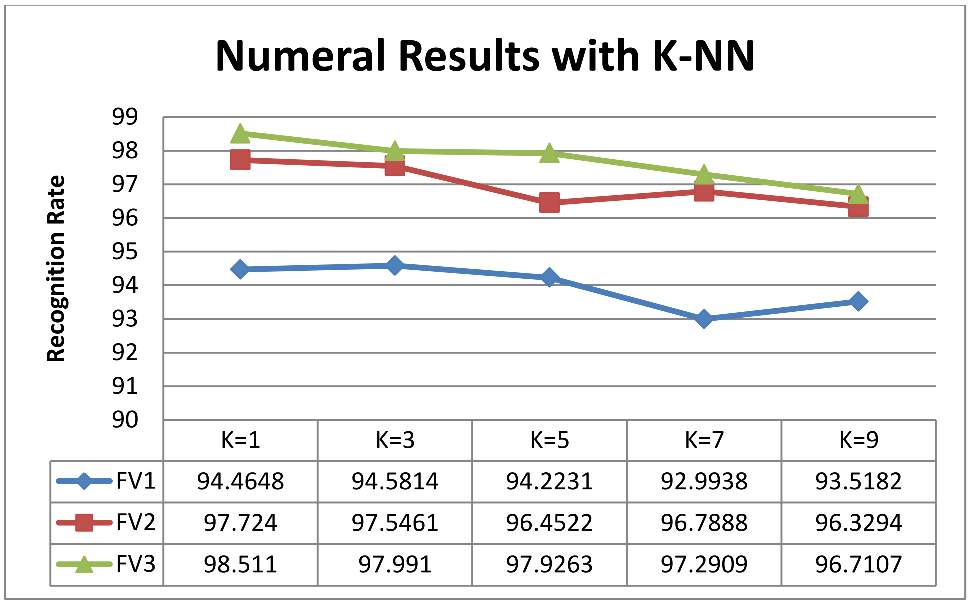
	γ = 0.0625	γ = 0.125	γ = 0.25	γ = 0.5	γ = 1	γ = 2
FV1	98.0667	98	97.7333	97.5333	93.7333	78.8
FV2	98.8667	99	99.2	99.2	98.5333	94.2667
FV3	98.8667	98.8667	99	99.1333	97.8	88.2667

Figure 4.2 Results with SVM at C = 16 and different values of γ



	σ = 0.05	σ = 0.10	σ = 0.15	σ = 0.20	σ = 0.25	σ = 0.30	σ = 0.35	σ = 0.40	σ = 0.45	σ = 0.50	σ = 0.55	σ = 0.60
FV1	89.3	94.3	95	95	95.7	96	95	94	93.3	93	91.7	90.7
FV2	96.7	96.7	97	97	97.3	98	97	96	95	94.3	94	93.7
FV3	98	98.3	98.3	98.3	97.7	98	98	98	97.7	97	96.3	96

Figure 4.3 Results of numeral recognition with PNN at different values of σ



	K=1	K=3	K=5	K=7	K=9
FV1	94.4648	94.5814	94.2231	92.9938	93.5182
FV2	97.724	97.5461	96.4522	96.7888	96.3294
FV3	98.511	97.991	97.9263	97.2909	96.7107

Figure 4.4 Results of numeral recognition with K-NN with different values of K

Confusion Matrix: In the table 4.5 confusion matrix for numeral recognition is given. The accuracy observed separately on different numerals is shown and the confusion with other numerals while recognition process is shown. It can be observed from the table that 2 is misclassified as 1 and 4; and 9 is misclassified as 8. This misclassification takes place because of similarity in the handwritten shapes of these Gurmukhi numerals.Because our test dataset used to recognize the Gurmukhi numerals is very small, so we couldn"t observe the misclassifications of other numerals.

Table 4.5 Confusion matrix of numeral recognition

Digit	Classified or misclassified as										Accuracy (%)
	1	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
0	27	0	0	0	0	0	0	0	0	0	100
1	0	29	0	0	0	0	0	0	0	0	100
2	0	1	33	0	1	0	0	0	0	0	94.29
3	0	0	0	35	0	0	0	0	0	0	100
4	0	0	0	0	31	0	0	0	0	0	100
5	0	0	0	0	0	32	0	0	0	0	100
6	0	0	0	0	0	0	22	0	0	0	100
7	0	0	0	0	0	0	0	24	0	0	100
8	0	0	0	0	0	0	0	0	34	0	100
9	0	0	0	0	0	0	0	0	1	30	96.77
Average accuracy											99

Discussion and Analysis

In Gurmukhi numeral recognition, The results obtained with SVM are highest comparative to other two classofiers- PNN and K-NN. We can conclude that we have obtained the maximum recognition rate using SVM classifier as 99.2% at 190 projection histogram features among all three feature sets. Secondly, we obtained 99.13% recognition rate at 144 zonal density and Background Directional Distribution features. Thirdly, we obtained 98.13% recognition rate at 128 distance profile features. It is clear here that feature set having large number of features is giving higher recognition rate. While recognizing with PNN and K-NN the highest results are obtained with FV3 feature set i.e. combination of zonal density and BDD feature.

Observing the SVM results, at the overall point of efficiency and performance, the recognition results using third feature set using zonal density and BDD features are best. It is because it has slight reduction in recognition rate (99.2% to 99.13%) but a significant reduction in number of features (190 to 144) comparative to second feature set. Hence it reduces computation complexity and hence time consumption while processing lesser number of features. In other way, comparing with first feature set, third features set provides significant increase in recognition rate (98% to 99.13%) while slight increase in number of features (128 to 144).

Chapter 5. CONCLUSION

In our thesis work we have recognized Gurmukhi characters and numerals both separately. We have used isolated Gurmukhi samples and numerals written on plain paper. Zonal density, distance profiles, projection histograms, and background directional distribution (BDD) features are used in different combinations to construct feature vectors. These feature vectors are used in SVM, K-NN and PNN classifiers for classification. The best recognition results are obtained with BDD features using SVM classifier and these results are enhanced when combined with other features.

Chapter 1 is about introduction and overview of OCR system, Gurmukhi script and our proposed work and methodology. This chapter presentsoverview of OCR phases including pre-processing, segmentation, feature extraction and classification processes in section 1.3. Chapter 2 is dedicated to our literature survey. The detailed description of our proposed work for Gurmukhi character recognition is given in chapter 3.We have also extended our methodology to recognize Gurmukhi numerals. Gurmukhi numeral recognition is described in chapter 4.

In our work first we collected handwritten samples from different writers. We pre-processed these samples of characters by applying many pre-processing techniques like median filtration, image dilation and some morphological operations to remove different types of irregularities and enhance the sample image. Generally practiced pre-processing approaches are briefly discussed in section 1.3.1, while the pre-processing techniques applied in our work are described in section 3.3, in particular.

Before extraction features the characters must be segmented into classifiable objects. Commonly these objects are individual characters or modifiers. In our case these are isolated basic characters of Gurmukhi scripts. Because our handwritten samples were collected in a separated writing of character, so our emphasis was not to go through detailed and complex segmentation. We simply segmented the characters by separating them on the basis of white space presented on paper in horizontal and vertical lines. A brief and general description about segmentation is presented in section 1.3.2.

After pre-processing and segmentation we have to extract the relevant features. Different methods of feature extraction generally used and many issues needed to consider while extracting features are described in section 1.3.3. The features extracted in our work and used to form 10 different sets of feature vectors are illustratively described in section 3.4. In particular to our work we have derived zonal density, distance profiles, projection histograms and background directional distribution features which are used to form 10 distinguished feature vectors. These feature vectors are used for classification of characters and numerals.

Character samples are classified in specified classes on the basis of constructed feature vectors with the help of classifiers. Different methods for classification process used generally in character recognition are described insection 1.3.4. In our work we have used SVM, K-NN and PNN classifiers. A description about these classifiers in theoretical and practical points of view is presented in section 3.5. In section 3.6 our observed results and analytical description arediscussed.

The best result for Gurmukhi character recognition is obtained with about 95% accuracy while the best result for numeral recognition is observed with about 99% accuracy. The best features derived are BDD and these features are enhanced by combining other features like zonal density, profile and histograms. In classifiers, best results are obtained with SVM. PNN and K-NN are at second and third place respectively in performance point of view. The most confusion among characters and numerals is arisen due to similar handwritten patterns. The confusion matrices for character and numeral recognition are provided in tables 3.5 and 4.8 respectively.

Future Scope

The work can be extended to increase the results by using or adding some more relevant features. We can use some features specific to the mostly confusing characters, to increase the recognition rate. We can divide the entire character set to apply specific and relevant features differently. More advanced classifiers as MQDF or MIL can be used and multiple classifiers can be combined to get better results.

Our work is constrained to isolated characters. To recognize strings in the form of words or sentences segmentation phase play a major role for segmentation at character level and modifier level. The situation becomes more critical, when the characters are joined, overlapped, and distorted. Hence there is major scope to extend the work to recognize such type of complex strings that will require advanced and complicated techniques mostly for segmentation phase.

To extend the work to string level recognition first line, word and character level will be required, then segmentation of characters into top, middle and bottom horizontal zones to separate upper and bottom modifiers will be required. Then these zoned objects can be recognized individually and these results can be placed in suitable context with other character or elements to form machine encoded text document equivalent to input document image.

REFERENCES

[1] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66, 1979

[2] Mehmet Sezgin and Bulent Sankur, “Survey over image thresholding techniques and quantitative performance evaluation”,*Journal of Electronic Imaging*Vol. 13, Issue 1, pp. 146–165, January 2004

[3] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, Ching Y. Suen, “Character Recognition Systems: A Guide for Students and Practioners”, *Wiley Inter-Science*, 2007

[4] Saiprakash Palakollu, Renu Shir, Rajneesh Rani, “Segmentation of Handwritten Devnagari Script”, *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 2, Issue 3, pp. 1244-1247, 2011

[5] O. D. Trier, A. K. Jain and T. Text, “Feature Extraction Methods For Character Recognition-A Survey”, *Pattern Recognition*, Vol. 29, No. 4, pp. 641-662, 1996

[6] Snehal Dalal and Latesh Malik, “A Survey of Methods and Strategies for Feature Extraction in Handwritten Script Identification”, *First International Conference on Emerging Trends in Engineering and Technology (ICETET)*, pp. 1164-1169,July 2008

[7] U. Pal, B.B. Chaudhury, “Indian Script Character Recognition: A Survey”, *Pattern Recognition, Elsevier*, pp. 1887-1899,2004

[8] G. Vamvakas, B. Gatos, S. Petridis, N. Stamatopoulos et al., “Optical Character Recognition for Handwritten Characters” ppt, [Online]. Available at: http://www.iit.demokritos.gr/IIT_SS/Presentations/Off-Line%20Handwritten%20OCR.ppt.

[9] Gurmukhi Script, Omniglot website [Online]. Available at: http://www.omniglot.com/writing/gurmuki.htm

[10] Gurmukhi Script, Sikh History website [Online]. Available at: http://www.sikh-history.com/sikhhist/events/gurmukhi.html

[11] Gurmukhi Alphabet – Introduction, Billie the Cat website [Online]. Available at: http://www.billie.grosse.is-a-geek.com/alphabet.html

[12] Gurmukhi Script, Wikipedia website [Online]. Available at: http://en.wikipedia.org/wiki/Gurmukh%C4%AB_script

[13] Online Punjabi Teaching Introduction, Learn Punjabi website. [Online]. Available at: http://www.learnpunjabi.org/intro1.asp

[14] Parzen Windows PDF, The University of Readingwebsite [Online]. Avaiable at: http://www.personal.reading.ac.uk/~sis01xh/teaching/CY2D2/Pattern2.pdf

[15] Parzen Window Density Estimation, The University of Utah website [Online]. Available at: http://www.cs.utah.edu/~suyash/Dissertation_html/node11.html

[16] Probabilistic Neural Network, The University of Reading Website [Online]. Available at: http://www.personal.reading.ac.uk/~sis01xh/teaching/CY2D2/Pattern3.pdf

[17] Frey Lab, PSI group, University of Toronto website [Online]. Available: http://www.psi.toronto.edu/~vincent/research/presentations/PNN.pdf

[18] Support Vector Machines (SVM), StatSoft website [Online]. Available at: http://www.statsoft.com/textbook/support-vector-machines/

[19] Support Vector Machine, Wikipedia website [Online]. Available at: http://en.wikipedia.org/wiki/Support_vector_machine

[20] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification", [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf , April 2010

[21] K-Nearest Neighbour Algorithm, Wikipedia website [Online]. Available at: http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[22] K-NN Classifiers, Phonetic Sciences Amsterdam website [Online]. Available at: http://www.fon.hum.uva.nl/praat/manual/kNN_classifiers_1__What_is_a_kNN_classifier_.html

[23] U. Pal, Wakabayashi, Kimura, "Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifiers", *10th International Conference on Document Analysis and Recognition*, pp. 1111-1115, 2009

[24] Vikas J Dungre et al., "A Review of Research on Devnagari Character Recognition", *International Journal of Computer Applications*, Volume-12, No.2, pp. 8-15, November 2010

[25] Sandhya Arora, D. Bhattacharjee, M. Nasipuri, D.K. Basu, M. Kundu, "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", *Industrial and Information Systems, IEEE Region 10 Colloquium and the Third ICIIS*, pp. 1-6, December, 2008

[26] S. Arora, D. Bhattacharjee, M. Nasipuri, M.Kundu, D.K. Basu, "Application of Statistical Features in Handwritten Devnagari Character Recognition", *International Journal of Recent Trends in Engineering* (IJRTE), Vol. 2, No. 2, pp. 40-42, November 2009

[27] V.L. Lajish,S.K. Kopparapu, "Fuzzy Directional Features for unconstrained on-line Devanagari handwriting recognition," *National Conference on Communications (NCC)*, pp.1-5, January 2010

[28] Prachi Mukherji, Priti Rege, "Shape Feature and Fuzzy Logic Based Offline Devnagari Handwritten Optical Character Recognition", *Journal of Pattern Recognition Research 4*,pp. 52-68, 2009

[29] G. Vamvakas, B. Gatos, S. Petridis, N. Stamatopoulos, "An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition", *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, Vol.2, pp. 1073-1077, September 2007

[30] Sarbajit Pal, Jhimli Mitra, Soumya Ghose, Paromita Banerjee, "A Projection Based Statistical Approach for Handwritten Character Recognition," *in Proceedings of International Conference on Computational Intelligence and Multimedia Applications*, Vol. 2, pp.404-408, 2007

[31] N. Araki, M. Okuzaki, Y. Konishi, H. Ishigaki, "A Statistical Approach for Handwritten Character Recognition Using Bayesian Filter" *3rd International Conference on Innovative Computing Information and Control (ICICIC)*, pp.194-198, June 2008

[32] Wang Jin, Tang Bin-bin, Piao Chang-hao, Lei Gai-hui, "Statistical method-based evolvable character recognition system",*IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 804-808, July 2009

[33] Apurva A. Desai, "Gujarati Handwritten Numeral Optical Character Reorganization through Neural Network", *Pattern Recognition*, Vol. 43, Issue 7, pp. 2582-89, July 2010

[34] D. Singh, S.K. Singh, M. Dutta, "Handwritten Characterr Recognition Using Twelve Directional Feature Input and Neural Network", *International Journal of Computer Application*s, Vol. 1 No.3, pp. 82-85,February 2010

[35] G. S. Lehal, C. Singh, "A Gurmukhi Script Recognition System", *Proceedings of 15th International Conference on Pattern Recognition*, Vol. 2, pp. 557-560, 2000

[36] G. S. Lehal, C. Singh, "A Complete Machine printed Gurmukhi OCR", *Vivek*, 2006

[37] G.S. Lehal, C. Singh, "Feature Extraction and Classification for OCR of Gurmukhi Script", *Vivek* Vol. 12, pp. 2-12, 1999

[38] G. S. Lehal, C. Singh, "A Post Processor for Gurmukhi OCR", *Sadhana*, Vol. 27, Part 1, pp. 99-111, 2002

[39] D. Sharma, G. S. Lehal, "An Iterative Algorithm for segmentation of Isolated Handwritten Words in Gurmukhi Script", *The 18th International Conference on Pattern Recognition (ICPR)*, Vol. 2, pp. 1022-1025, 2006

[40] V. Goyal, G.S. Lehal, "Comparative Study of Hindi and Punjabi Language Scripts", *Nepalese Linguistics*, Vol. 23, pp. 67-82, 2008

[41] G.S. Lehal, Nivedan Bhatt, "A Recognition System for Devnagari and English Handwritten Numerals", *Proc. ICMI, Springer*, pp. 442-449, 2000

[42] D. Sharma, G. S. Lehal, Preety Kathuria, "Digit Extraction and Recognition from Machine Printed Gurmukhi Documents", *Proceedings of the International Workshop on Multilingual OCR MORC Spain*, 2009

[43] Anuj Sharma, Rajesh Kumar, R. K. Sharma, "Online Handwritten Gurmukhi Character Recognition Using Elastic Matching",*Conference onImage and Signal Processing (CISP)*, Vol.2, pp.391-396, May 2008

[44] Anuj Sharma, R.K. Sharma, Rajesh Kumar, "Online Handwritten Gurmukhi Character Recognition", Ph.D. Thesis, Thapar University, 2009 [Online]. Available: http://dspace.thapar.edu:8080/dspace/bitstream/10266/1057/3/Thesis_AnujSharma_SMCA_9041451.pdf

[45] Naveen Garg, Karun Verma, “Handwritten Gurmukhi Charcter Recognition Using Neural Network”, M.Tech. Theis, Thapar University, 2009 [online]. Available: http://dspace.thapar.edu:8080/dspace/bitstream/10266/788/1/thesis+report+final.pdf

[46] Puneet Jhajj, D. Sharma, “Recognition of Isolated Handwritten Characters in Gurmukhi Script”, *International Journal of Computer Applications*, Vol. 4, No. 8, pp. 9-17, August 2010

[47] Ubeeka Jain, D. Sharma, “Recognition of Isolated Handwritten Characters of Gurumukhi Script using Neocognitron”, *International Journal of Computer Applications*, Vol. 4, No. 8, pp. 10-16, November 2010

[48] Kartar Siddharth, Renu Dhir, Rajneesh Rani, “Handwritten Gurumukhi Charater Recognition Using Zoning Density and Background Directional Features”, *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 2, Issue 3, pp. 1036-1041, May-June 2011

[49] Kartar Singh Sidharth, Mahesh Jangid, Renu Dhir, Rajneesh Rani, “Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features”, *International Journal of Computer Science and Engineering (IJCSE)*, Vol. 3, No. 6, pp. 2332-2345, June 2011

[50] Kartar Singh Siddharth, Renu Dhir, Rajneesh Rani, “Handwritten Gurmukhi Numeral Recognition Using Different Feature Sets”, *Communicated in International Journal*

[51] Reena Bajaj, Lipika Dey, Shantanu Chaudhuri, “Devnagari numeral recognition by combining decision of multiple connectionist classifiers”, *Sadhana* Vol. 27, Part 1, pp. 59–72, February 2002

[52] U. Bhattacharya, B. B. Chaudhuri, "A Majority Voting Scheme for Multiresolution Recognition of Handprinted Numerals”, *Seventh International Conference on Document Analysis and Recognition (ICDAR) –* Vol. 1, pp. 16, 2003

[53] U. Bhattacharya, S.K. Parui, B. Shaw, K. Bhattacharya, “Neural Combination of ANN and HMM for Handwritten Devanagari Numeral Recognition”, *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006

[54] U. Pal, T. Wakabayashi, N. Sharma, F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts," *International conference on Document Analysis and Recognition (ICDAR)*, Vol.2, pp.749-753, 2007

[55] P.M. Patil, T.R. Sontakke, “Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network”, *Pattern Recognition, Elsevier*, Vol. 40, Issue 7, pp. 2110-2117, July 2007

[56] R.J. Ramteke, S.C. Mehrotra, “Recognition of Handwritten Devanagari Numerals”, *International Journal of Computer Processing of Oriental Languages, Chinese Language Computer Society & World Scientific Publishing Company*, pp. 1-9, 2008

[57] Shailedra Kumar Shrivastava, Sanjay S. Gharde, “Support Vector Machine for Handwritten Devanagari Numeral Recognition”, *International Journal of Computer Applications*, Vol. 7, No. 11, pp. 9-14, October 2010

[58] Mahesh Jangid, Kartar Singh, Renu Dhir, Rajneesh Rani, “Performance Comparison of Devanagari Handwritten Numerals Recognition”, *Internation Journal of Computer Applications*, Vol. 22, No.1, pp. 1-6, May 2011

[59] G.G. Rajput, S.M. Mali, “Fourier Descriptor Based Isolated Marathi Handwritten Numeral Recognition”, *International Journal of Computer Applications*, Vol. 3, No. 4, pp. 9-13, June 2010

[60] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[61] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, “A Practical Guide to Support Vector Classification”, [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

LIST OF PUBLICATIONS

[1]. Kartar Siddharth, Renu Dhir, Rajneesh Rani, "Handwritten Gurumukhi Charater Recognition Using Zoning Density and Background Directional Features", *International Journal of Computer Science and Information Technologies (IJCSIT)*, Vol. 2, Issue 3, pp. 1036-1041, May-June 2011

[2]. Kartar Singh Sidharth, Mahesh Jangid, Renu Dhir, Rajneesh Rani, "Handwritten Gurmukhi Character Recognition Using Statistical and Background Directional Distribution Features", *International Journal of Computer Science and Engineering (IJCSE)*, Vol. 3, No. 6, pp. 2332-2345, June 2011

[3]. Kartar Singh Siddharth, Renu Dhir, Rajneesh Rani, "Handwritten Gurmukhi Numeral Recognition Using Different Feature Sets", *International Journal of Computer Applications (IJCA),* Volume 28, No. 2, pp. 20-24, August 2011

[4]. Kartar Singh Siddharth, Renu Dhir, Rajneesh Rani, "Comparative Recognition of Handwritten Gurmukhi Numerals Using Different Feature Sets and Classifiers", *(Communicated)*